505-41-30

EOSDIS Core System Project

Interface Control Document Between the EOSDIS Core System (ECS) and the Version 0 System for Interoperability

Revision A

June 1996

National Aeronautics and
Space Administration

GODDARD SPACE FLIGHT C
GREENBELT, MARYLAND

INTERFACE CONTROL DOCUMENT
between the
EOSDIS Core System (ECS) and the
Version 0 System for Interoperability

Reviewed by:

_____	_____
Greg Hunolt DAAC System/Science Operations Manager GSFC - Code 505	Date
_____	_____
Candace Carlisle Interface Manager GSFC - Code 505	Date
_____	_____
Melvin Banks Chief, ESDIS Development GSFC - Code 505	Date

Approved by:

_____	_____
Dale Harris Associate Director for ESDIS GSFC - Code 505	Date

GODDARD SPACE FLIGHT CENTER

GREENBELT, MARYLAND

This page intentionally left blank.

Preface

This document is a formal contract deliverable with an approval code 1. It requires Government review and approval prior to acceptance and use. This document is under ECS contractor configuration control. Once this document is approved, Contractor approved changes are handled in accordance with Class I and Class II change control requirements described in the EOS Configuration Management Plan, and changes to this document shall be made by document change notice (DCN) or by complete revision.

Any questions should be addressed to:

ESDIS Project Configuration Management Office
Code 505
Goddard Space Flight Center NASA
Greenbelt, MD 20771

This page intentionally left blank.

Abstract

The Distributed Active Archive Centers (DAACs) support the science community by supplying, via ECS and the EOSDIS Version 0 (V0) Systems, data archive, distribution, information management and product generation services for a wide range of data sets related to global change research. At Release A and Release B, Level 3 two-way catalog interoperability between ECS and the EOSDIS V0 Systems will enable users of either system to search, browse, and order data products made available by the other system. This Interface Control Document (ICD) specifically defines the design of each catalog interoperability interface between ECS and the EOSDIS V0 Information Management System (IMS).

Specifically, this ICD defines the data flows that exist between ECS and the EOSDIS V0 IMS for two-way catalog interoperability, including the following: directory search request/results, guide search request/results, inventory search request/results, browse request/results, product request/result, acknowledge, pong, statistics, quit, and dependent valids.

On the V0 side of the interface, some of these messages are implemented using Object Description Language (ODL); while others use HTTP GET commands and WAIS queries. Data flows/exchanges involving these protocols are identified as follows:

- V0 System users accessing ECS services interface directly through the EOSDIS V0 IMS Client to ECS at the DAACs. The EOSDIS V0 IMS Client interfaces with the ECS Document Data Server (Guide) using WAIS formatted queries over HTTP protocol. Also, using ODL, the EOSDIS V0 IMS Client interfaces, via the V0 Gateway (a part of ECS), to the ECS Science Data Server. The V0 Gateway translates the V0 user's ODL service request into Object Oriented Distributed Computing Environment (OODCE); in addition, at Release B, Illustra's version of SQL is used as the Earth Science Query Language (ESQL) for ECS.

- (Release A only) ECS users accessing V0 System services interface directly through the ECS Release A Search and Order Tool (RASOT), which is, in fact, a reused copy of the EOSDIS V0 IMS Client, Release 6.0. The ECS RASOT uses ODL to talk to the EOSDIS V0 IMS Servers, and HTTP and WAIS to talk to the V0 DAAC Guide Servers.

- (For Release B only) ECS users accessing V0 System services interface directly through the ECS ESST. Using OODCE/ESQL, the ECS ESST interfaces via the V0 Gateway to both the V0 DAAC Guide Servers and the EOSDIS V0 IMS Servers. To accommodate the interface to the V0 DAAC Guide Servers, the V0 Gateway first translates the ECS user's service request into HTTP and WAIS. To accommodate the interface to the EOSDIS V0 IMS Servers, the V0 Gateway first translates the ECS user's service request into ODL.

This ICD is consistent with the ECS-to-V0 System interface requirements, as described in the Earth Science Data and Information System (ESDIS) Project -- Level 2 Requirements, the Functional and Performance Requirements Specification for the Earth Observing System Data and Information System (EOSDIS) Core System (ECS Level 3 requirements), and the Interface Requirements Document (IRD) Between EOSDIS Core System (ECS) and the V0 System.

Keywords: active, archive, center, bi-directional, browse, catalog, control, DAAC, directory, distributed, Earth, ESST, FTP, granule, guide, HTTP, interface, interoperability, inventory, ODL, object, order, product, RASOT, release, science, search, system, tool, TRMM, valids, Version 0, Version 1, V0, V1, WAIS

Change Information Page

ISSUE	DATE	PAGES AFFECTED	DESCRIPTION
Baseline	05/20/96	All	CCR 505-41-37-002
Revision A	06/24/96	All	CCR-505-41-30-002-B

This page intentionally left blank.

List of Affected Pages

Page No.	Revision	Page No.	Revision	Page No.	Revision	Page No.	Revision
Title	Revision A	3-4	Revision A	4-22	Revision A	5-8	Revision A
i	Revision A	3-5	Revision A	4-23	Revision A	5-9	Revision A
ii	Revision A	3-6	Revision A	4-24	Revision A	5-10	Revision A
iii	Revision A	4-1	Revision A	4-25	Revision A	5-11	Revision A
iv	Revision A	4-2	Revision A	4-26	Revision A	5-12	Revision A
v	Revision A	4-3	Revision A	4-27	Revision A	5-13	Revision A
vi	Revision A	4-4	Revision A	4-28	Revision A	5-14	Revision A
vii	Revision A	4-5	Revision A	4-29	Revision A	5-15	Revision A
viii	Revision A	4-6	Revision A	4-30	Revision A	5-16	Revision A
ix	Revision A	4-7	Revision A	4-31	Revision A	5-17	Revision A
x	Revision A	4-8	Revision A	4-32	Revision A	5-18	Revision A
xi	Revision A	4-9	Revision A	4-33	Revision A	5-19	Revision A
xii	Revision A	4-10	Revision A	4-34	Revision A	5-20	Revision A
xiii	Revision A	4-11	Revision A	4-35	Revision A	5-21	Revision A
xiv	Revision A	4-12	Revision A	4-36	Revision A	5-22	Revision A
1-1	Revision A	4-13	Revision A	4-37	Revision A	A-1	Revision A
1-2	Revision A	4-14	Revision A	4-38	Revision A	A-2	Revision A
2-1	Revision A	4-15	Revision A	5-1	Revision A	A-3	Revision A
2-2	Revision A	4-16	Revision A	5-2	Revision A	A-4	Revision A
2-3	Revision A	4-17	Revision A	5-3	Revision A	A-5	Revision A
2-4	Revision A	4-18	Revision A	5-4	Revision A	A-6	Revision A
3-1	Revision A	4-19	Revision A	5-5	Revision A	A-7	Revision A
3-2	Revision A	4-20	Revision A	5-6	Revision A	A-8	Revision A
3-3	Revision A	4-21	Revision A	5-7	Revision A	A-9	Revision A

List of Affected Pages

Page No.	Revision	Page No.	Revision	Page No.	Revision	Page No.	Revision
A-10	Revision A						
A-11	Revision A						
A-12	Revision A						
A-13	Revision A						
A-14	Revision A						
A-15	Revision A						
A-16	Revision A						
AB-1	Revision A						
AB-2	Revision A						

Contents

Figures

Appendix A. Dependent Valids Format

Abbreviations and Acronyms

This page intentionally left blank.

1. Introduction

1.1 Identification

This Interface Control Document (ICD), Contract Data Requirements List (CDRL) Item 029 whose requirements are specified in Data Item Description (DID) 209/SE1, is a required deliverable under the Earth Observing System Data and Information System (EOSDIS) Core System (ECS) Contract (NAS5-60000).

1.2 Scope

This Interface Control Document (ICD) defines the system interfaces that exist between ECS and the EOSDIS V0 System Information Management System (IMS) for Level 3 catalog interoperability. ECS Releases are keyed to mission support: Release Ir-1 provides support to TRMM Early Interface Testing and Science Algorithm I&T. Release A provides support to TRMM Science Operations and TRMM Ground Systems Certification Testing. Release A also provides the functional capabilities needed to support early ESDIS Ground System Testing for the EOS AM-1 and Landsat 7 missions. Release B provides support to EOS AM-1 Mission Operations and Science Operations, and it provides support to ESDIS Ground System Certification Testing for the EOS AM-1 and Landsat 7 missions. Release B also provides archive and distribution services for the Landsat 7 mission. Releases C & D provide evolutionary enhancements to the ECS services provided in the earlier Releases.

This ICD does not address internetworking for V0-to-ECS catalog interoperability--this topic is addressed in each of the ECS to DAAC ICDs. Furthermore, interfaces for the migration of V0 data sets are not addressed herein, but are documented in the Version 1 Data Migration Plan.

The Earth Science Data and Information System (ESDIS) Project has responsibility for the development and maintenance of this ICD. Any changes in the interface requirements must be agreed to and assessed at the ESDIS Project Level. This ICD will be approved under the signature of the ESDIS Project Manager.

This document reflects the technical baseline, maintained by the ECS Configuration Control Board in accordance with ECS technical direction (see Section 2.2).

1.3 Purpose and Objectives

This document is written to formalize the interpretation and general understanding of the interfaces between ECS and the V0 System IMS. This document provides a point of mutual control of external interface definitions via the ESDIS Configuration Control Board (CCB).

1.4 Status and Schedule

This is the final ICD for the ECS-V0 System catalog interoperability interfaces which will be implemented in ECS. This ICD has been submitted as an ECS Project CCB approval Code 1 document. At the Government's option, this document may be designated to be under full Government CCB control. Changes may be submitted for consideration by Contractor and Government CCBs under the normal change process at any time.

1.5 Organization

Section 1 provides information regarding the identification, scope, purpose and objectives, and organization of this document.

Section 2 provides a listing of the related documents, which were used as a source of information for this document.

Section 3 provides an overview of the two-way catalog interoperability interfaces between the ECS and the Version 0 System. Specifically, this section describes the purpose of the catalog interoperability interfaces and a high level description of the data flows (as an introduction to the detailed information provided in Section 4). This section also includes a context diagram.

Section 4 includes the following information:

- A definition of the Object Description Language (ODL) Message Normalization Form (i.e., Group Structure) is provided for each applicable message. In addition, a brief discussion of the ODL conventions is provided in the context of an example.
- The detailed definition of each data interface (i.e., message exchanged) between the EOSDIS V0 IMS Client and the ECS Science Data Server (or ECS Document Data Server)
- (Release A only) The detailed definition of each data interface between the ECS Release A Search and Order Tool (RASOT) and the EOSDIS V0 IMS Server (or V0 DAAC Guide Servers)
- (Release B only) The detailed definition of each data interface between the ECS Earth Science Search Tool (ESST) and the EOSDIS V0 IMS Server (or V0 DAAC Guide Servers), via the V0 Gateway

Section 5 provides a detailed definition of each of the ODL keywords corresponding to the ODL Message Normalization Forms identified in Section 4.

Appendix A contains the dependent valids format.

Abbreviations and Acronyms contains an acronym list.

2. Related Documents

2.1 Parent Documents

The following are parent documents from which this document's scope and content derive:

193-208-SE1-001	Methodology for Definition of External Interfaces for the ECS Project
301-CD-002-003	System Implementation Plan for the ECS Project
423-10-01-1	Goddard Space Flight Center, Earth Science Data and Information System (ESDIS) Project Level 2 Requirements, EOSDIS Core System (ECS), Volume 1, 1/27/93
423-10-01-5	Goddard Space Flight Center, Earth Science Data and Information System (ESDIS) Project Level 2 Requirements, Volume 5: EOSDIS Version 0; through CH-01, 9/13/93
423-41-01	Goddard Space Flight Center, EOSDIS Core System (ECS) Statement of Work, through CN-14, 4/25/95
423-41-02	Goddard Space Flight Center, Functional and Performance Requirements Specification for the Earth Observing System Data and Information System (EOSDIS) Core System (ECS) Revision A through CH-06, 4/25/95
505-10-20	System Interface Control Plan for the Earth Science Data and Information System (ESDIS Project)
505-41-11	Goddard Space Flight Center, Interface Requirements Document Between EOSDIS Core System (ECS) and the Version 0 System, 10/95

2.2 Applicable Documents

The following documents are referenced herein and are directly applicable to this document. In the event of conflict between any of these documents and this ICD, this document shall take precedence. Please note that Internet links cannot be guaranteed for accuracy or currency.

305-CD-023-002	Release B SDPS Data Management Subsystem Design Specification for the ECS Project
313-CD-003-002	Communications and System Management Segment (CSMS) Internal Interface Control Document for the ECS Project, Preliminary
210-TP-001-006	Technical Baseline for the ECS Project

209-CD-008-004	Interface Control Document (ICD) Between the EOSDIS Core System (ECS) and the Goddard Space Flight Center (GSFC) Distributed Active Archive Center (DAAC) for the ECS Project
209-CD-010-003	Interface Control Document (ICD) Between the EOSDIS Core System (ECS) and the Langley Research Center (LaRC) Distributed Active Archive Center (DAAC) for the ECS Project
209-CD-021-002	Interface Control Document (ICD) Between the EOSDIS Core System (ECS) and the Alaska SAR (Synthetic Aperture Radar) Facility (ASF) Distributed Active Archive Center (DAAC) for the ECS Project
209-CD-022-002	Interface Control Document Between EOSDIS Core System (ECS) and the Oak Ridge National Laboratory (ORNL) Distributed Active Archive Center (DAAC) for the ECS Project
IMSV0-PD-SD-002	Hughes STX Corporation, Messages and Development Data Dictionary-V0 and Release A Message Passing Protocol Specification, 9/95
IMSV0-SW-DE-003	Hughes STX Corporation, EOSDIS IMS Guide Subsystem Design Document, 10/93
540-032	Goddard Space Flight Center, EBnet - Distributed Active Archive Center (DAAC) Interface Control Document (ICD)
none	Hughes STX Corporation, IMS Server Cookbook: Setting Up An IMS Server, undated
none	Hughes, STX Corporation, An Overview of Valids Support File Processing in V0 IMS Version 6 and the Release A Search and Order Tool
none	Goddard Space Flight Center, ECS Technical Direction No. 11, "PDR Technical Baseline," 12/6/94
none	Davis, Randy; University of Colorado Laboratory for Atmospheric and Space Physics: User's Guide for the Object Description Language (ODL) Processing Software Library, Release 2.1 DRAFT, 3/13/91
none	Planetary Data System Standards Reference, Version 3.1, 8/94 (WWW access: http://stardust.jpl.nasa.gov/stdref/stdref.html)

2.3 Information Documents

The following documents, although not directly applicable, amplify or clarify the information presented in this document, but are not binding.

604-CD-001-004	Operations Concept for the ECS Project: Part 1-- ECS Overview
604-CD-002-003	Operations Concept for the ECS project: Part 2B -- Release B
604-CD-003-002	Operations Concept for the ECS Project: Part 2A -- Release A
160-TP-002-001	Version 1 Data Migration Plan, Technical Paper
441-TP-001-001	Implementation Plan for the Release A Client, Technical Paper

This page intentionally left blank.

3. Interface Overview

The interface between the ECS and the V0 System IMS supports Level 3 two-way catalog interoperability to provide an exchange of data and information. Specifically, this interface supports the search, location and acquisition of data between ECS and the V0 System, providing ECS and V0 System users with ready access to the data and services provided by the other system.

Figures 3-1 and 3-2 display high level context diagrams for the catalog interoperability interfaces between ECS and the V0 System at, respectively, Releases A and B. The catalog interoperability interfaces supported are categorized as follows:

a. directory search request/results – for finding datasets

b. guide search request/results – for obtaining detailed information about datasets

c. inventory search request/results – for locating specific granules within a dataset

d. acknowledge – to acknowledge reception of inventory search results chunk

e. browse requests/results – for enabling the user to retrieve/view "representative" images, as well as non-image data. There are two browse modes available, including

   (1) FTP browse – in response to a browse request submitted through the client, "representative" images are automatically staged at a designated FTP site, and a message is returned to the user containing sufficient information to allow the user to retrieve the images via FTP

   (2) integrated browse – in response to a browse request, "representative" images are returned directly to the user via the client for viewing

f. product requests/results – placement of orders for full resolution data sets

g. quit – notification of premature termination of a session due to problems; also used at the normal termination of inventory results exchanges of chunks.

h. pong – used to test for server availability and responsiveness (not supported by ESST).

i. statistics – client session statistics which are sent to the server (not supported by ESST).

On the V0 side of the interface, some of these messages are implemented using Object Description Language (ODL); while others use HTTP GET commands and WAIS queries. Data flows/exchanges involving these protocols are identified as follows (specific messages utilizing HTTP, WAIS or ODL, are identified and described in Section 4):

- V0 System users accessing ECS services interface directly through the EOSDIS V0 IMS Client to ECS at the DAACs. The EOSDIS V0 IMS Client interfaces directly with the

ECS Document Data Server (Guide) using WAIS formatted queries over HTTP protocol. Also, using ODL, the EOSDIS V0 IMS Client interfaces, via the V0 Gateway (a part of ECS), to the ECS Science Data Server. The V0 Gateway translates the V0 user's ODL service request into Object Oriented Distributed Computing Environment (OODCE); in addition, at Release B, Illustra's version of SQL is used as the Earth Science Query Language (ESQL) for ECS.

- (Release A only) ECS users accessing V0 System services interface directly through the ECS RASOT, which is, in fact, a reused copy of the EOSDIS V0 IMS Client, Release 6.0. The ECS RASOT uses ODL to talk to the EOSDIS V0 IMS Servers, and HTTP and WAIS to talk to the V0 DAAC Guide Servers.
- (For Release B only) ECS users accessing V0 System services interface directly through the ECS ESST. Using OODCE/ESQL, the ECS ESST interfaces via the V0 Gateway to both the V0 DAAC Guide Servers and the EOSDIS V0 IMS Servers. To accommodate the interface to the V0 DAAC Guide Servers, the V0 Gateway first translates the ECS user's service request into HTTP and WAIS. To accommodate the interface to the EOSDIS V0 IMS Servers, the V0 Gateway first translates the ECS user's service request into ODL.

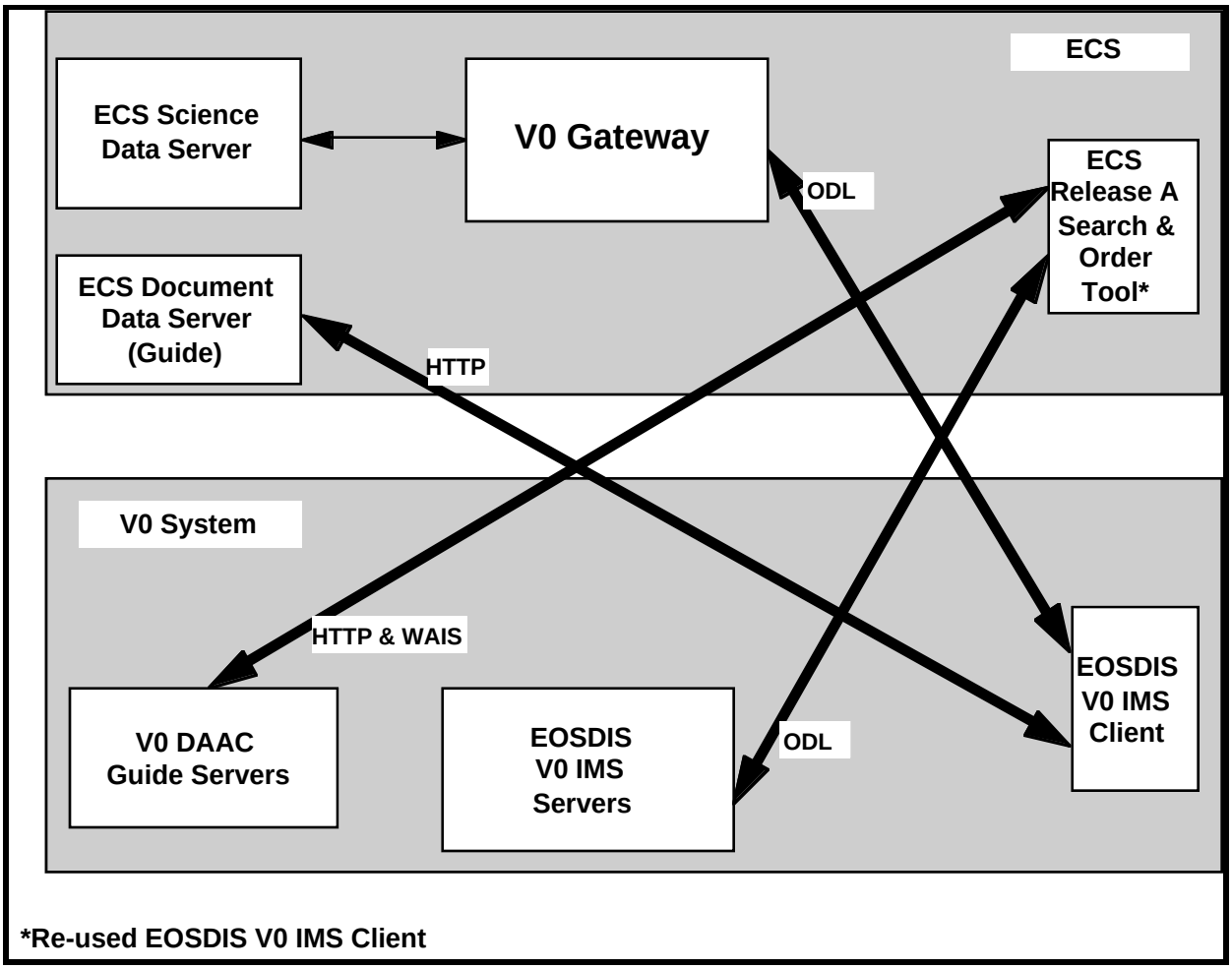


Figure 3-1. ECS/V0 System Interoperability Context Diagram (Release A)

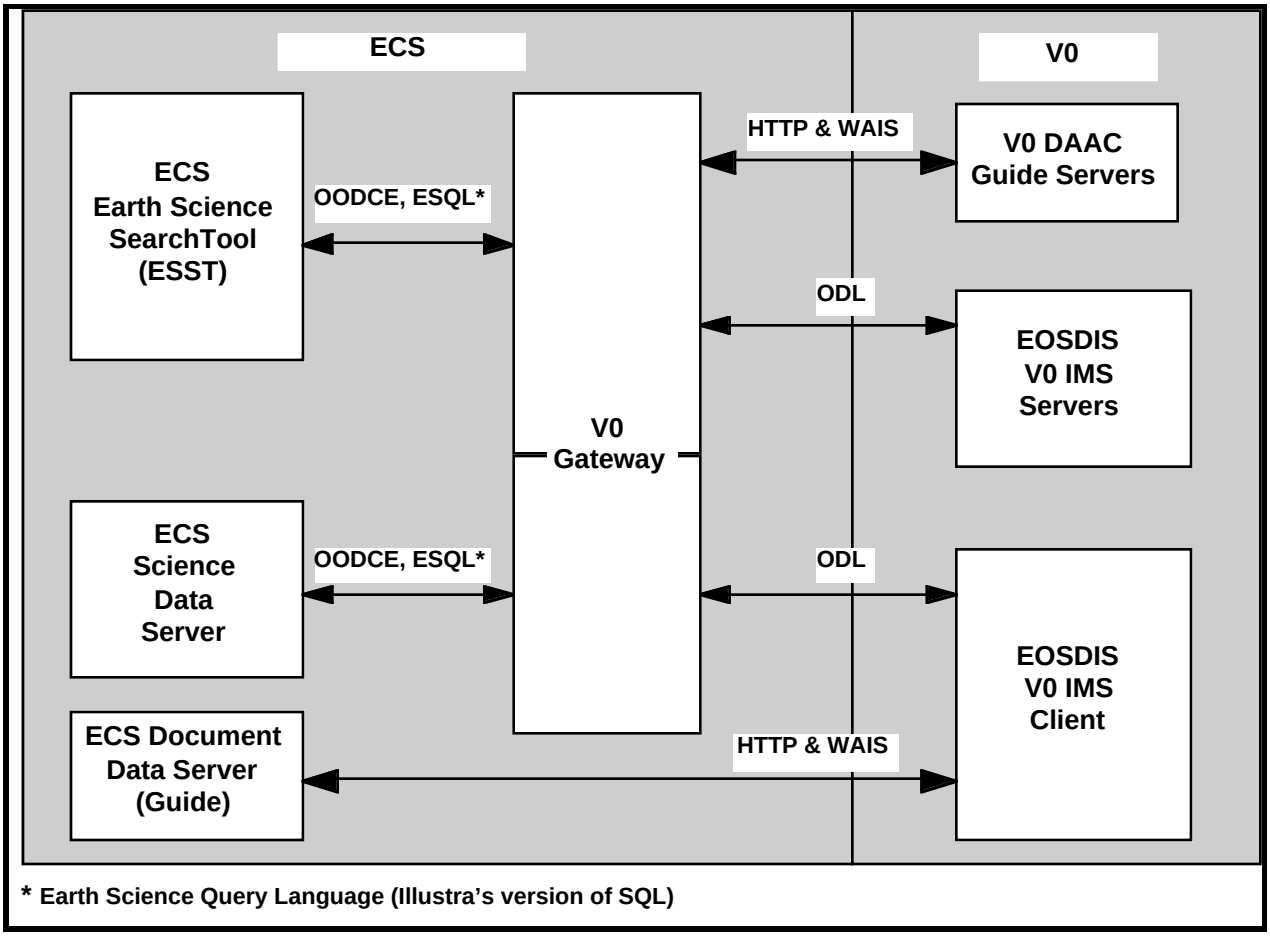


Figure 3-2. ECS/V0 System Interoperability Context Diagram (Release B)

In order to help users form meaningful queries to the IMS, valid values for the various fields are provided through what are called "dependent valids." A list of the possible values which are consistent across all archives is prepared by the V0/ECS Science Team (i.e., system-level team consisting of both V0 and ECS personnel) and presented to the user for selection. Once the user has selected one or more values in one search field, displays in other search fields present only those values that are compatible with the previously selected values. These dependencies are maintained in a set of support files that are automatically updated whenever the IMS is run.

Each DAAC maintains all valid search field values, and interdependencies between them for its datasets, and manually supplies that information in flat ASCII files using ODL syntax to the V0/ECS Science Team. The valids submitted by the V0 DAACs/ECS are compared against a master file maintained by the V0/ECS Science Team. Any new valids are noted and passed to the V0/ECS Science Team, which checks that the new terms are consistent with the existing body of terminology being used by the V0/ECS tools. If there are any inconsistencies or problems, negotiation between the V0/ECS Science Team and the V0 DAACs/ECS would be required. Once negotiation is done, the valids support files are built in a bit-mapped format (see Appendix A for details) by an ingest program from the information provided by the V0 DAACs/ECS.

(Release A only) At client start-up, the latest valids are loaded into the EOSDIS V0 IMS Client or ECS RASOT from a "valids support-file server" that is run at at least two sites for redundancy.

(Release B only) At client start-up, the latest valids are loaded into the EOSDIS V0 IMS Client or, via the V0 Gateway, into the ECS Earth Science Search Tool (ESST) from a "valids support-file server" that is run at at least two sites for redundancy.

More detailed information on valids support file processing can be found in the document, "An Overview of Valids Support File Processing in V0 IMS Version 6 and the Release A Search and Order Tool."

Also, for information pertaining to internetworking/communications for V0/ECS interoperability, the reader should refer to the following documents:

- #305-CD-028-002, Release B CSMS Communication Subsystem Design Specification for the ECS Project
- #540-032, Goddard Space Flight Center, EBnet - Distributed Active Archive Center (DAAC) Interface Control Document (ICD)
- #209-CD-008-004, Interface Control Document Between the EOSDIS Core System (ECS) and the Goddard Space Flight Center (GSFC) Distributed Active Archive Center (DAAC) for the ECS Project.
- #209-CD-010-003, Interface Control Document Between the EOSDIS Core System (ECS) and the Langley Research Center (LaRC) Distributed Active Archive Center (DAAC) for the ECS Project.

- #209-CD-021-002, Interface Control Document (ICD) Between the EOSDIS Core System (ECS) and the Alaska SAR (Synthetic Aperture Radar) Facility (ASF) Distributed Active Archive Center (DAAC) for the ECS Project.
- #209-CD-022-002, Interface Control Document Between EOSDIS Core System (ECS) and the Oak Ridge National Laboratory (ORNL) Distributed Active Archive Center (DAAC) for the ECS Project

4. Data Flow Descriptions

4.1 General

This section contains the detailed definition of each data interface between ECS and the EOSDIS V0 System IMS (i.e., Client and Server) that is required to support two-way catalog interoperability. In particular, an identification of each data flow is provided along with a discussion of the functional purpose of that flow and the detailed format and contents of each interface. For discussion purposes, this section treats the two-way interoperability data flows as two separate one-way interoperability data flows, including the following:

- data flows between EOSDIS V0 IMS Client and ECS Servers, and
- either
  - (Release A only) data flows between ECS RASOT and V0 IMS Servers, or
  - (Release B only) data flows between ECS ESST (via the V0 Gateway) and V0 IMS Servers

Since many of the above-referenced messages are implemented with ODL, an example of the ODL normalization forms and standardized conventions is provided in Figure 4-1. These standardized conventions, which provide a formal method of describing ODL commands, include the following rules:

- keywords are words that have a special meaning in ODL, itself, and are treated as instructions.
- all keyword are printed in CAPS
- items in square brackets ([ ]) are options.
- items in parentheses (...) indicate that these items may be repeated any number of times
- after the parentheses (...) a single character is given that tells how many occurrences are allowed; i.e.,
  - a '*' means zero or more occurrences
  - a ‘+’ means one or more occurrences
- Each group is further defined down to its keyword components.

In Section 5, each keyword is defined in terms of the following items of information, as appropriate:

- synopsis (short English-Language description of the keyword),
- parent groups,
- children,
- ODL type; e.g.,
  - integer,
  - real,
  - date,
  - string,
  - aggregate,
  - symbol,
  - sequence string,
  - character string
- maximum (value) length
- possible values.

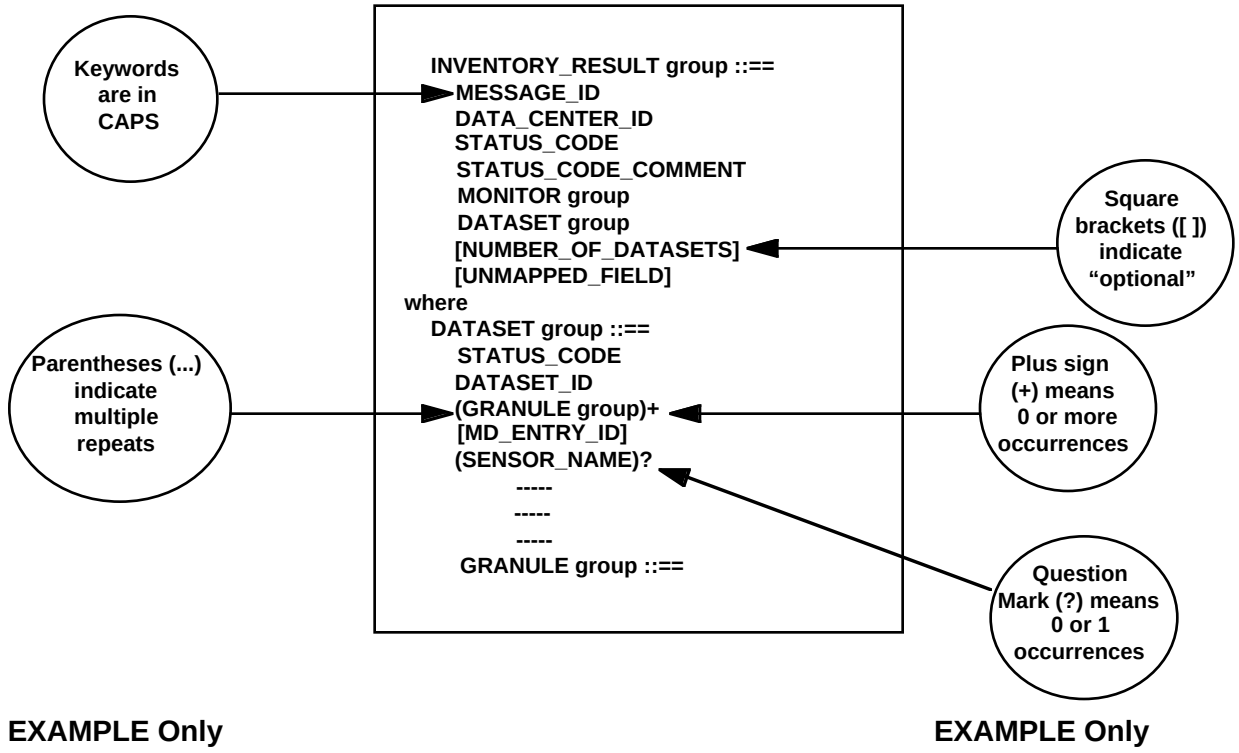


Figure 4-1. Example of ODL Normalization Form Illustrating Conventions

4.2 Data Flows Between ECS RASOT or ECS ESST (via V0 Gateway) and V0 IMS Servers

The data flows between the ECS RASOT and the V0 IMS Servers (at Release A) and between the ECS ESST (via the V0 Gateway) and the V0 IMS Servers (at Release B) are discussed in this section and are depicted, respectively, in Figures 4-2 and 4-3 (it should be noted that data flows between the EOSDIS V0 IMS Client and the ECS Servers are discussed in section 4.3). Specifically, the following data flows are represented:

- (Release A only) Between the ECS RASOT and the EOSDIS V0 IMS Server:
  - Directory Search Request
  - Directory Search Results
  - Inventory Search Request

  - Inventory Search Results
  - Acknowledge
  - Browse Request
  - FTP Browse Results
  - Integrated Browse Results
  - Product Request
  - Product Results
  - Quit
  - Statistics
  - Pong
- (Release A only) Between the ECS RASOT and the V0 DAAC Guide Servers:
  - Guide Search Request
  - Guide Search Results
- (Release B only) Between the ECS ESST (via the V0 Gateway) and the EOSDIS V0 IMS Server:
  - Directory Search Request
  - Directory Search Results
  - Inventory Search Request
  - Inventory Search Results
  - Acknowledge
  - Browse Request
  - FTP Browse Results
  - Integrated Browse Results
  - Product Request
  - Product Results
  - Quit
- (Release B only) Between the ECS ESST (via the V0 Gateway) and the V0 DAAC Guide Servers
  - Guide Search Request

- Guide Search Results

With the exception of the guide search request/results messages, all messages use Object Description Language (ODL). (For a description of ODL refer to the User's Guide for the Object Description Language Processing Software Library, Release 2.1 - Draft.) All of these messages are handled by the IMS Kernel (IK) layer [Note: The ECS RASOT, EOSDIS V0 IMS Server and the V0 Gateway contain several software modules, at the communications (lowest) layer, which serve as library routines and are, collectively, referred to as the IK layer]. Each of these messages is described, in detail, in the sections which follow.

(Release A only) Between the ECS RASOT and the V0 DAAC Guide Servers, WAIS formatted queries over a modified version of WAIS 0.5 protocol are used for guide request/results. [It should be noted that the WAIS 0.5 server has been modified to consider "=" and "_" as alphabetic characters.]

(Release B only) Between the V0 Gateway and the V0 DAAC Guide Servers, WAIS formatted queries over a modified version of WAIS 0.5 protocol are used for guide request/results. [It should be noted that the WAIS 0.5 server has been modified to consider "=" and "_" as alphabetic characters.] The V0 Gateway translates between this V0 protocol and OODCE/ESQL.

(Release A only) Pong and statistics messages are invoked by the ECS RASOT and transmitted by the "Ponger" program (an executable that is separate from, but spawned by, the ECS RASOT) to the EOSDIS V0 IMS Server.

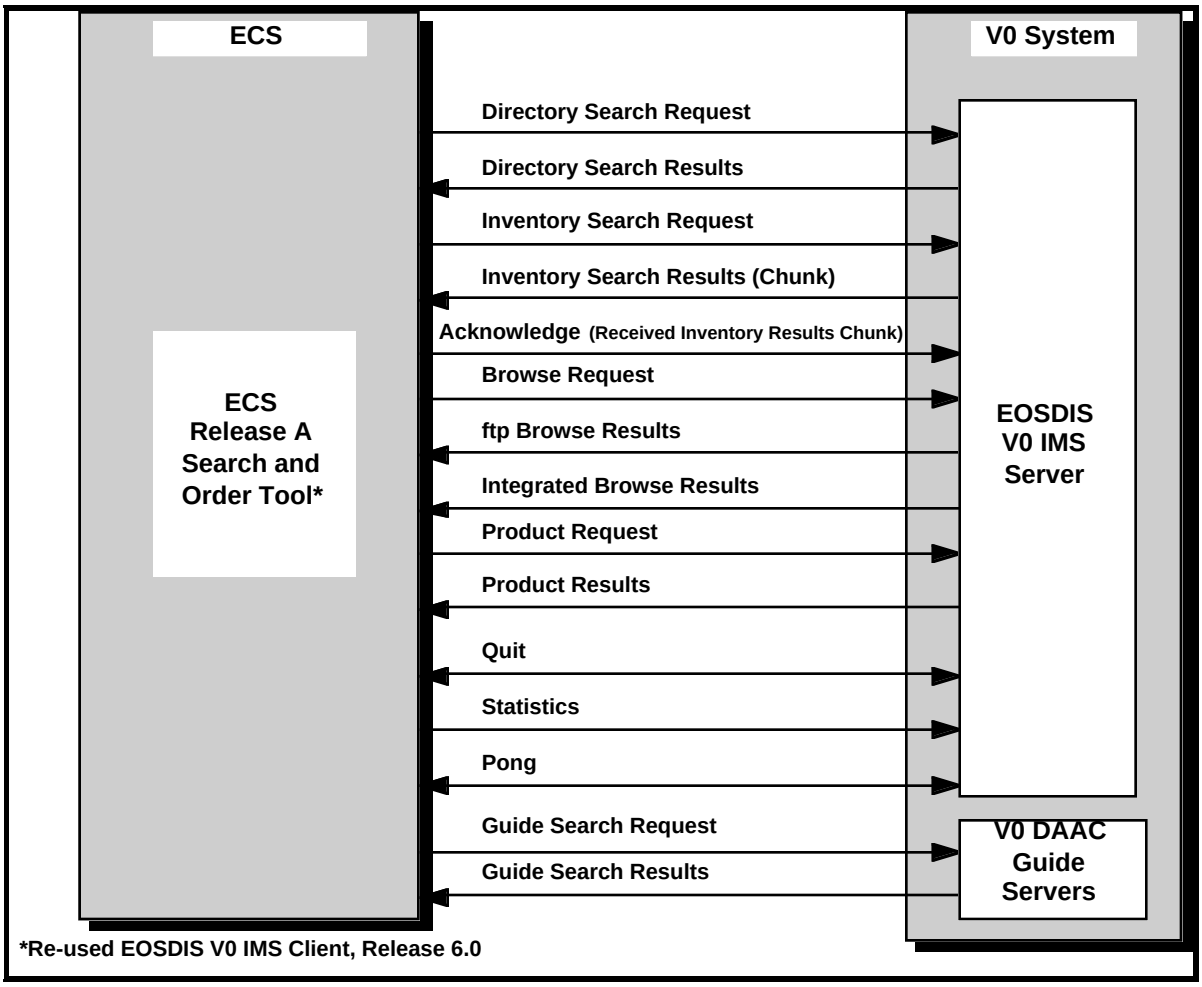


Figure 4-2. Interfaces Between ECS RASOT and V0 IMS Servers (Release A)

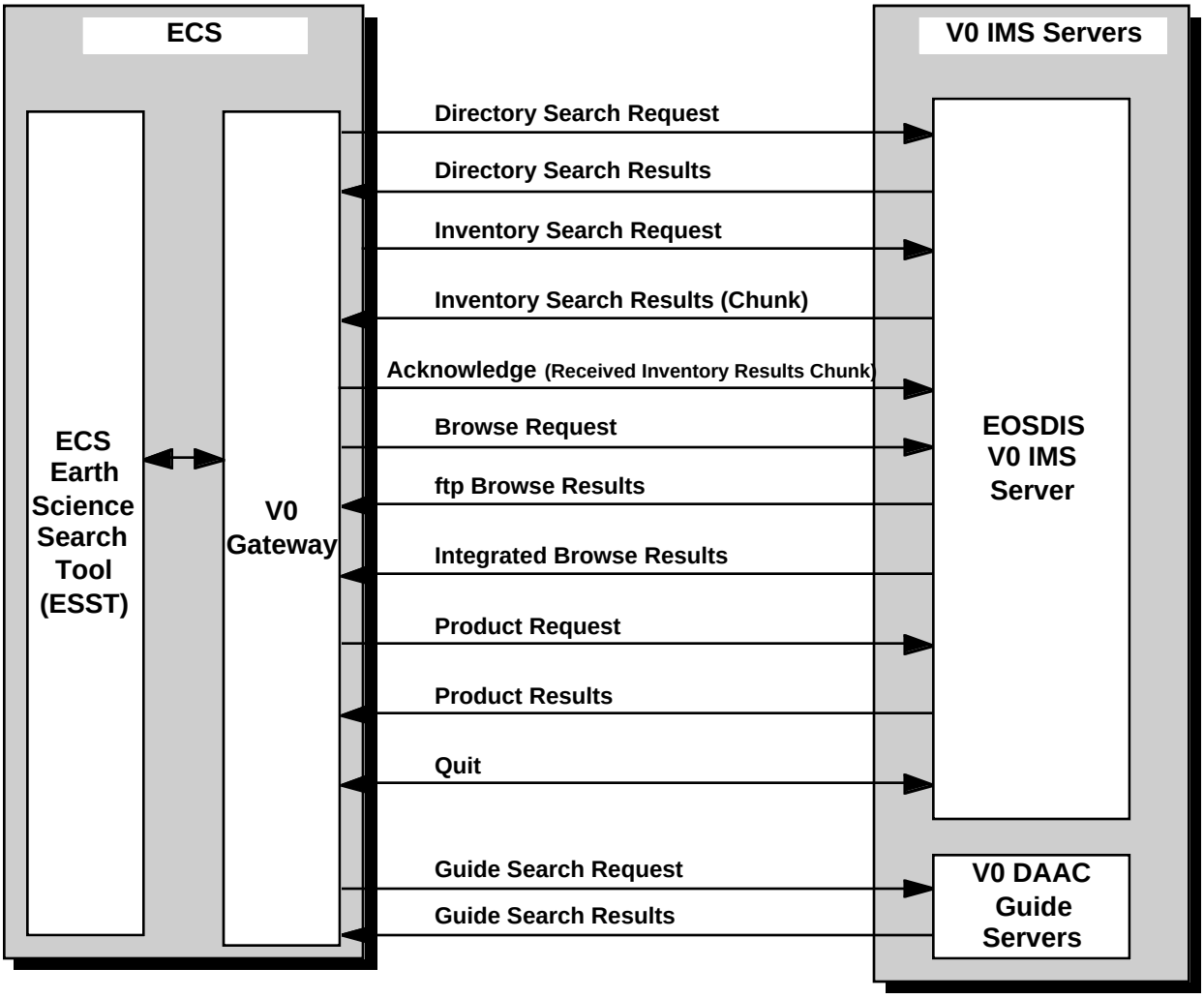


Figure 4-3. Interfaces Between ECS ESST (via V0 Gateway) and V0 IMS Servers (Release B)

4.2.1 Directory Search Request/Results

The purpose of the directory search is to aid the user in making an initial determination of the potential usefulness of various data sets pertinent to some application by searching through descriptions of metadata or data set catalogues which contain high-level information. The directory search provides information on the location of metadata or data set catalogues. The search criteria, specified by the user, are based on the following typical searchable attributes: source, sensor, geophysical parameter, dataset name, data center id, geographical coordinates (area), temporal intervals, etc.

(Release A only) An ECS user, requesting V0 System services, submits the directory search request via the ECS RASOT. The ECS RASOT sends the request to the EOSDIS V0 IMS Server.

The EOSDIS V0 IMS Server returns Global Change Master Directory (GCMD) Entry Identifiers (i.e., keyword MD_ENTRY_ID) to the ECS RASOT. The ECS RASOT accesses the GCMD using these MD_ENTRY_ID keywords. The GCMD then returns directory information to the ECS RASOT.

(Release B only) An ECS user, requesting V0 System services, submits the directory search request via the ECS ESST. The ECS ESST sends the request, via the V0 Gateway, to the EOSDIS V0 IMS Server. The EOSDIS V0 IMS Server returns, via the V0 Gateway, Global Change Master Directory (GCMD) Entry Identifiers (i.e., keyword MD_ENTRY_ID) to the ECS ESST. The ECS ESST accesses the GCMD using these MD_ENTRY_ID keywords. The GCMD then returns directory information to the ECS ESST.

The Directory Search Request and Directory Search Results messages are implemented using ODL---their ODL Normalization Forms are defined in the immediately-following sections. [A discussion of the ODL standardized conventions is provided as reference in Section 4.1. Detailed definitions of the message keywords (e.g., MESSAGE_ID) are provided in Section 5.]

4.2.1.1 ODL Normalization Form for Directory Search Request

The ODL Normalization Form for directory search request defined in this section pertains, in its entirety, to the following interfaces:

- (Release A only) ECS RASOT-to-EOSDIS V0 IMS Server
- (Release B only) ECS ESST (via the V0 Gateway)-to-EOSDIS V0 IMS Server

```
DIRECTORY_SEARCH group ::==
    MESSAGE_ID
    [AUTHENTICATOR]
    [ECS_AUTHENTICATOR]
    MONITOR group
    [CAMPAIGN]
    [DATASET_ID]
    [PARAMETER]
    [SENSOR_NAME]
    [SOURCE_NAME]
    [START_DATE]
    [STOP_DATE]
    [RANGE_LOC group]
    VERSION group

RANGE_LOC group ::==
    NORTH_LATITUDE
    SOUTH_LATITUDE
    EAST_LONGITUDE
    WEST_LONGITUDE
```

MONITOR group ::==
TX_CLIENT
[RX_SERVER]
[TX_SERVER]
[RX_CLIENT]

VERSION group ::==
PROTOCOL_VERSION
SENDER_VERSION
[IMS_STAFF]

4.2.1.2 ODL Normalization Form for Directory Search Results

The ODL Normalization form for directory search results defined in this section pertains, in its entirety, to the following interfaces:

- (Release A only) EOSDIS V0 IMS Server-to-ECS RASOT
- (Release B only) EOSDIS V0 IMS Server (via the V0 Gateway)-to-ECS ESST

DIRECTORY_RESULT group ::==
MESSAGE_ID
DATA_CENTER_ID
STATUS_CODE
[STATUS_CODE_COMMENT]
(DATASET group)+
NUMBER_OF_DATASETS
MONITOR group
VERSION group

DATASET group ::==
DATASET_ID
MD_ENTRY_ID
[ORG_CENTER]

MONITOR group ::==
TX_CLIENT
RX_SERVER
TX_SERVER
[RX_CLIENT]

VERSION group ::==
PROTOCOL_VERSION
SENDER_VERSION

[IMS_STAFF]

4.2.2 Guide Search Request/Results

The purpose of the guide search is to locate and retrieve guide documents containing detailed information about datasets, based on user-specified keywords or freetext strings; it also allows a user to locate datasets if (s)he decides not to go through a directory or inventory search first. In addition, guide serves to aid a user in understanding the metadata in the search construction process. The search criteria, specified by the user, are based on the following typical searchable attributes: source, sensor, geophysical parameter, dataset name, data center id, geographical coordinates (area), temporal intervals, etc. An ECS user can look through the V0 guide documents, follow any hypertext link and display any related documents. The guide documents give detailed information on campaign, source, sensor, data_center, and dataset_id.

4.2.2.1 Guide Search Request

(Release A only) The ECS RASOT sends search requests for documents as WAIS formatted queries over a modified version of WAIS 0.5 to the V0 DAAC Guide Servers. [It should be noted that, specifically, WAIS 0.5 was modified to consider "+" and "_" as alphabetic characters, so that it will index a document under (for example) “PARAMETER=BRIGHTNESS_TEMPERATURE” as a single “word.” It is these keyword=value "words" that the guide search uses to find its documents.] The valids processing in the ECS RASOT determines which guide servers to send the guide query to, and then creates a separate search message for each destination server based on the information in the archive.odl file (depicted in Figure 4-4). The archive.odl file configured with each ECS RASOT identifies the V0 DAAC Guide Servers as WAIS servers---specifically, Figure 4-4 depicts an example of the archive.odl file documenting the server address and WAIS protocol for connecting to a V0 DAAC Guide Server (bold emphasis added). The parameters and values in the guide search queries are of the same domain as the search and order parameters and values used for the Directory and Inventory search functions in the ECS RASOT. The WAIS servers at the V0 DAACs receive the guide queries and perform the search using a WAIS index.

(Release B only) The ECS ESST sends (via the V0 Gateway) search requests for documents in OODCE/ESQL. The V0 Gateway converts this OODCE/ESQL into WAIS formatted queries over a modified version of WAIS 0.5 and sends these queries to the V0 DAAC Guide Servers. [It should be noted that, specifically, WAIS 0.5 was modified to consider "+" and "_" as alphabetic characters, so that it will index a document under (for example) “PARAMETER=BRIGHTNESS_TEMPERATURE” as a single “word.” It is these keyword=value "words" that the guide search uses to find its documents.] The valids processing in the V0 Gateway determines which guide servers to send the guide query to, and then creates a separate search message for each destination server based on the information in the archive.odl file (depicted in Figure 4-4). The archive.odl file configured with the V0 Gateway identifies the V0 DAAC Guide Servers as WAIS servers---specifically, Figure 4-4 depicts an example of the archive.odl file documenting the server address and WAIS protocol for connecting to a V0 DAAC Guide Server (bold emphasis added). The parameters and values in the guide search queries are of the same domain as the search and order parameters and values used for the Directory and

Inventory search functions in the ECS ESST. The WAIS servers at the V0 DAACs receive the guide queries and perform the search using a WAIS index.

```
/* $Id: archive.odl,v 4.3.4.1 1995/08/02 17:08:26 ims Exp $ */

/* OPERATIONAL/STABLE/DEMO archive information */

GROUP              = DATA_CENTER_INFO
  GROUP       = DATA_CENTER
     DATA_CENTER_ID             = "ASF"
     DATA_CENTER_NAME    = "ALASKA SAR FACILITY"
     INTERNET           = "eosims.asf.alaska.edu"
     PORT               = "12325"
    GUIDE_SRV_ADDR      = "wais://eosims.asf.alaska.edu:12365/ASF_guide"
  END_GROUP   = DATA_CENTER
                                        o
                                        o
```

Figure 4-4. Example of archive.odl File Documenting Server Address and WAIS Protocol for Connecting to a V0 DAAC Guide Server (bold emphasis added).

4.2.2.2 Guide Search Results

(Release A only) Each of the V0 DAAC Guide Servers creates a results "hit" list formatted using HTML, and sends the HTML page over a modified WAIS 0.5 connection, independently of the other V0 DAAC Guide Servers. The ECS RASOT does not integrate the results of the multiple V0 DAAC Guide Servers into a merged list. Each document "hit" is displayed as a hyperlink in the guide results window in the ECS RASOT. When the user selects a guide document for viewing, the request is sent as an HTTP message using the HTTP protocol. The HTTP message contains the path name of the document and the server address which stores the document. The document is returned to the ECS RASOT over the HTTP connection.

(Release B only) Each of the V0 DAAC Guide Servers creates a results "hit" list formatted using HTML, and sends the HTML page over a modified WAIS 0.5 connection to the V0 Gateway, independently of the other V0 DAAC Guide Servers. The V0 Gateway converts this V0 protocol into OODCE/ESQL. The ECS ESST does not integrate the results of the multiple V0 DAAC Guide Servers into a merged list. Each document "hit" is displayed as a hyperlink in the guide results window in the ECS ESST. When the user selects a guide document for viewing, the request is sent from the ESST to the V0 Gateway where it is converted to an HTTP message using HTTP protocol, and sent to the V0 DAAC Guide Servers. The HTTP message contains the path name of the document and the server address which stores the document. The document is returned to the ECS ESST via the V0 Gateway.

4.2.3 Inventory Search Request/Results and Acknowledge

The purpose of the inventory search is to aid a user in searching through the available inventory, locating and retrieving metadata about specific granules of the product(s) of interest, and determining whether any granules should be ordered; and also to allow a user to find datasets if the user chooses not to use a directory or guide search first. The search criteria, specified by the user, are based on the following typical searchable attributes: source, sensor, geophysical parameter, dataset name, data center id, geographical coordinates (area), temporal intervals, etc.

(Release A only) An ECS user, requesting V0 System services, submits the inventory search request via the ECS RASOT. The ECS RASOT sends the EOSDIS V0 IMS Server inventory search criteria based on characteristics of the data. The EOSDIS V0 IMS Server retrieves the requested granules' metadata, and sends these items back to the ECS RASOT.

(Release B only) An ECS user, requesting V0 System services, submits the inventory search request via the ECS ESST. The ECS ESST, via the V0 Gateway, sends the EOSDIS V0 IMS Server inventory search criteria based on characteristics of the data. The EOSDIS V0 IMS Server retrieves the requested granules' metadata, and sends these items back, via the V0 Gateway, to the ECS ESST.

In cases where the Inventory Search Results are large, an Inventory Results message can be broken up, according to a set of rules, into "chunks" which are smaller but complete trees---the size of chunks need not be uniform, but should be moderate in size. The basic "building blocks" for a chunk/tree include the following items of information (see Section 4.2.3.2):

a. Inventory Result Prefix – This item of information consists of the following sub-items:

   1. Message_ID
   2. Data_Center_ID
   3. Status_Code
   4. Status_Code_Comment (optional)
   5. Unmapped_Field (optional)

   According to the rule, every chunk/tree must contain an Inventory Result Prefix.

b. Package Group - This includes metadata about collections of granules that can be ordered from an archive. The package group can be part of a dataset group or can be outside the dataset groups according to three options to be discussed in the paragraphs below.

c. Dataset Group – This item includes metadata within the Dataset Group. Every chunk may contain 0 or more items of Dataset Group metadata.

d. Granule Group – This item includes metadata within the Granule Group. According to the rule, every chunk will include 0 or more Granule Group information items. It is always part of a dataset group.

A package is a collection of granules or data which can be ordered from an archive. An EOSDIS V0 IMS Server can integrate package information into the chunk/tree according to the following

three options:

- Option 1 – Insert all Package Groups ahead of the first Dataset Group
- Option 2 – Insert relevant Package Groups ahead of each Dataset Group
- Option 3 – Embed relevant Package Groups inside each Dataset Group

As a result of the integration of the package information into chunks/trees according to these three options, the following combinations of information are permitted for chunks:

- Chunk = Inventory Result Prefix followed by Package Information
- Chunk = Inventory Result Prefix + Package Information followed by Dataset Metadata (possibly containing Granule information)
- Chunk = Inventory Result Prefix followed by Dataset metadata (usually containing Granules information)
- Chunk = Inventory Result Prefix followed by Dataset metadata (containing Package Information and possibly Granules information)

(Release A only) The ECS RASOT returns a separate acknowledge message to the EOSDIS V0 IMS Server upon receiving each chunk.

(Release B only) The ECS ESST, via the V0 Gateway, returns a separate acknowledge message to the EOSDIS V0 IMS Server upon receiving each chunk.

The Inventory Search Request and Inventory Search Results messages are implemented using ODL---their ODL Normalization Forms are defined in the immediately-following sections. [A discussion of the ODL standardized conventions is provided as reference in Section 4.1. Detailed definitions of the message keywords (e.g., MESSAGE_ID) are provided in Section 5.]

(Release B Only) In order to accommodate two-way mapping of terminology between ECS and the V0 System, the V0 Gateway (a part of ECS) maintains a Sybase database containing the terminology mapping information. The V0 Gateway database is built by a Gateway Administrator using V0 System search parameters, ECS schema and metadata. Specifically, upon receiving a request from the ECS ESST, the V0 Gateway performs a ECS-V0 mapping table look-up within the V0 Gateway database, converting the ECS request into V0's terminology in order

to accommodate the EOSDIS V0 IMS Servers. Similarly, results returned from the EOSDIS V0 IMS Servers to the V0 Gateway are converted, via the V0-ECS mapping service, to ECS terminology prior to returning these results to the ECS ESST. The V0 Gateway-to-Sybase mapping interfaces are internal to ECS, and are completely documented in CDRL #305-CD-023-002, Release B SDPS Data Management Subsystem Design Specification for the ECS Project.

4.2.3.1 ODL Normalization Form for Inventory Search Request

The ODL Normalization Form for inventory search request defined in this section pertains, in its entirety, to the following interfaces:

- (Release A only) ECS RASOT-to-EOSDIS V0 IMS Server
- (Release B only) ECS ESST (via the V0 Gateway)-to-EOSDIS V0 IMS Server

INVENTORY_SEARCH group ::==
    MESSAGE_ID
    [AUTHENTICATOR]
    [ECS_AUTHENTICATOR]
    GRANULE_LIMIT
    [BROWSE_ONLY]
    [CAMPAIGN]
    [DATASET_ID]
    [SENSOR_NAME]
    [SOURCE_NAME]
    [START_DATE]
    [STOP_DATE]
    [START_DAY_OF_YEAR]
    [STOP_DAY_OF_YEAR]
    [DAY_NIGHT]
    [PROCESSING_LEVEL]
    [PARAMETER]
    GLOBAL_GRANULES_ONLY ‖
    POINT_LOC group ‖
    POLYGON_LOC group ‖
    RANGE_LOC group ‖
    XHAIRS group ‖
    MONITOR group
    VERSION group

Note:
One of these five groups must be sent with the search (based on user selection).

POINT_LOC group ::==
    LATITUDE
    LONGITUDE

POLYGON_LOC group ::==
    LATITUDE
    LONGITUDE
    [POLE_INCLUDED]
    MAP_PROJECTION_TYPE
    TANGENT_LATITUDE
    TANGENT_LONGITUDE

RANGE_LOC group ::==
    NORTH_LATITUDE
    SOUTH_LATITUDE
    EAST_LONGITUDE
    WEST_LONGITUDE

XHAIRS group ::==
    LATITUDE
    LONGITUDE

LATITUDE_DISTANCE
LONGITUDE_DISTANCE

MONITOR group ::==
TX_CLIENT
[RX_SERVER]
[TX_SERVER]
[RX_CLIENT]

VERSION group ::==
PROTOCOL_VERSION
SENDER_VERSION
[IMS_STAFF]

4.2.3.2 ODL Normalization Form for Inventory Search Results

The ODL Normalization Form for inventory search results defined in this section pertains, in its entirety, to the following interfaces:

- (Release A only) EOSDIS V0 IMS Server-to-ECS RASOT
- (Release B only) EOSDIS V0 IMS Server (via the V0 Gateway)-to-ECS ESST

Note: Source, sensor and parameter information can be put either in DATASET or GRANULE groups. See annotations.

INVENTORY_RESULT group ::==
MESSAGE_ID
DATA_CENTER_ID
STATUS_CODE
[STATUS_CODE_COMMENT]
MONITOR group
(PACKAGE group)* repeated group
OPTION 1: for use when all package information is sent for the whole inventory result.
OPTION 2: for use when package information is sent in front of each relevant dataset group.
(DATASET group)*
[NUMBER_OF_DATASETS] (present only in the last chunk for an inventory results set)
[UNMAPPED_FIELD]

PACKAGE group ::==
DATA_CENTER_ID
DATASET_ID
PACKAGE_ID The PACKAGE_ID in the PACKAGE group gives an arbitrary identifier by which the package is known. Processing and media options for the package are provided in the group. GRANULE groups can list multiple packages in which they are available. For

the common case where granules can be ordered in single-granule packages and all such packages have the same processing and media options, a single package group can be provided whose id is "*". Then each granule that can be ordered this way can be listed as being in PACKAGE_ID "*" (along with possibly other named packages).
COMMENT
[INFO_PROMPT]
NUMBER_OF_GRANULES
NUMBER_OF_OPTIONS
(PROCESSING_OPTIONS group)+

(MEDIA_TYPE group)+

DATASET group ::==
STATUS_CODE
DATASET_ID
(VALID_ACCOUNTS group)* repeated group
(PACKAGE group)* ::== repeated group
OPTION 3: for use when package information is sent within each relevant dataset group and before the granule group(s).
(GRANULE group)* repeated group
[MD_ENTRY_ID]
[SENSOR_NAME] If all granules of the dataset have the same values for SENSOR_NAME, the value can be specified in the DATASET group and omitted from all of the GRANULE groups.)
[SOURCE_NAME] If all granules of the dataset have the same values for SOURCE_NAME, the value can be specified in the DATASET group and omitted from all of the GRANULE groups.)
[PARAMETER] If all granules of the dataset have the same values for PARAMETER_NAME, the value can be specified in the DATASET group and omitted from all of the GRANULE groups.)
[COMMENT]
[RESTRICTION]
[CAMPAIGN]
[DAY_NIGHT]
[PROCESSING_LEVEL]
[NUMBER_OF_GRANULE_HITS] (omitted from all chunks except the one containing the last granule of the dataset)
[BROWSE_PRODUCT_DESCRIPTION] (the headings should be done in UPPERCASE on lines by themselves in the sequence, i.e. PRIMARY PURPOSE, PRODUCT HISTORY, etc.)

VALID_ACCOUNTS group ::==
ACCOUNT_NUMBER
[BALANCE]
[ERROR]

GRANULE group ::==
GRANULE_ID
START_DATE
STOP_DATE

[SENSOR_NAME] If all granules of the dataset have the same values for SENSOR_NAME, the value can be specified in the DATASET group and omitted from all of the GRANULE groups.)
[SOURCE_NAME] If all granules of the dataset have the same values for SOURCE_NAME, the value can be specified in the DATASET group and omitted from all of the GRANULE groups.)
[PARAMETER] If all granules of the dataset have the same values for PARAMETER_NAME, the value can be specified in the DATASET group and omitted from all of the GRANULE groups.)
[BROWSE_TYPE]
[CAMPAIGN]
[COMMENT]
[DAY_NIGHT]
[PROCESSING_LEVEL]
[PACKAGE_ID] (If omitted or if package information is not provided within the inventory results, granule cannot be ordered.)
GLOBAL_GRANULE
POINT_LOC group |
POLYGON_LOC group |
RANGE_LOC group

POINT_LOC group ::==
LATITUDE
LONGITUDE

POLYGON_LOC group ::==
LATITUDE
LONGITUDE
[POLE_INCLUDED]
CENTROID_LAT
CENTROID_LON

RANGE_LOC group ::==
NORTH_LATITUDE
SOUTH_LATITUDE
EAST_LONGITUDE
WEST_LONGITUDE

PROCESSING_OPTIONS group ::==
    OPTION_ID
    PACKAGE_SIZE
    NUMBER_OF_MEDIA_TYPE
    (MEDIA_TYPE group)+

MEDIA_TYPE group ::==
    TYPE_ID
    NUMBER_OF_MEDIA_FORMAT
    (MEDIA_FORMAT)+

MEDIA_FORMAT group ::==
    FORMAT_ID
    APPROX_COST

MONITOR group ::==
    TX_CLIENT
    RX_SERVER
    TX_SERVER
    [RX_CLIENT]

4.2.3.3 ODL Normalization Form for Acknowledge

The ODL Normalization Form for the acknowledge message defined in this section pertains, in its entirety, to the following interfaces:

- (Release A only) ECS RASOT-to-EOSDIS V0 IMS Server
- (Release B only) ECS ESST (via the V0 Gateway)-to-EOSDIS V0 IMS Server

ACKNOWLEDGE group ::==
    MESSAGE_ID
    MONITOR group
    VERSION group

MONITOR group ::==
    TX_CLIENT
    [RX_SERVER]
    [TX_SERVER]
    [RX_CLIENT]

VERSION group ::==
    PROTOCOL_VERSION
    SENDER_VERSION
    [IMS_STAFF]

4.2.4 Browse Request/Results

The purpose of the Browse service is to allow the user to request and receive "representative" images for FTP-copying or viewing and for analysis prior to deciding on specific full-resolution products to order. Two browse service modes are available to a user, including FTP browse and integrated browse.

(Release A only) The FTP Browse service allows the user to FTP-copy the browse product onto the user's system. After the EOSDIS V0 IMS Server receives an FTP browse request (BROWSE_TYPE = FTP_Only), it transmits the FTP Browse Results to the ECS RASOT and copies the browse product onto the DAAC FTP site.

(Release B only) The FTP Browse service allows the user to FTP-copy the browse product onto the user's system. After the EOSDIS V0 IMS Server receives an FTP browse request (BROWSE_TYPE = FTP_Only), it transmits the FTP Browse Results, via the V0 Gateway, to the ECS ESST, and copies the browse product onto the DAAC FTP site.

The user is notified, via email that the browse product is ready to FTP. This notification provides the access information for the FTP including the following: (1) FTP account, (2) IP address of the FTP account, (3) directory and file name for each of the requested browse products, (4) pick up expiration date of staged files. The FTP browse product remains on the FTP site for an operations-tunable time interval, giving the user time to copy it onto the user's system.

(Release A only) The Integrated Browse service allows the user to view the browse product through the ECS RASOT. In response to an integrated browse request (BROWSE_TYPE = Y) sent by the ECS RASOT to the EOSDIS V0 IMS Server, the EOSDIS V0 IMS Server sends back, to the ECS RASOT, the integrated browse results message, followed by the browse image, which is then displayed to the user.

(Release B only) The Integrated Browse service allows the user to view the browse product through the ECS ESST. In response to an integrated browse request (BROWSE_TYPE = Y) sent by the ECS ESST, via the V0 Gateway, to the EOSDIS V0 IMS Server, the EOSDIS V0 IMS Server sends back, via the V0 Gateway, to the ECS ESST, the integrated browse results message, followed by the browse image which is then displayed to the user.

All V0 browse images (FTP and integrated) are provided in the National Super Computing Applications (NCSA) Hierarchical Data Format (HDF), Version 4.0.

The Browse Request/Results messages are implemented using ODL---their ODL Normalization Forms are defined in the immediately-following sections. [A discussion of the ODL standardized conventions is provided as a reference in Section 4.1. Detailed definitions of the message keywords (e.g., MESSAGE_ID) are provided in Section 5.]

EOSView is an interactive software tool which is used to aid a user in the selection, verification, presentation, and analysis of browse data files written in HDF or HDF-EOS format, Version 4.0 (also backward-compatible with earlier versions)---it should be noted that HDF-EOS is merely an

extension of HDF. Specifically, EOSView is designed to help the user to interactively visualize HDF or HDF-EOS browse data files during the selection of data, to verify that the data received is the data desired, and to get data which resides in some of the more common visualization and analysis systems, such as interactive data language (IDL).

(At Release A) EOSView can be run either concurrently with the ECS RASOT, or as a standalone application. The ECS RASOT, on its own, can display V0 image layers of HDF browse data files---it can also save the file to a user-selectable directory for viewing by EOSView or other viewers. EOSView also supports viewing FTP browse on the user's workstation.

(At Release B) EOSView can be run either as part of the ECS ESST, or as a standalone application. EOSView is integrated with the ECS ESST, and is initiated via scripts. From EOSView, looking at a V0 browse image in HDF format, the ECS user is able to specify that he/she wants to acquire this data. EOSView will communicate this back to the ECS ESST. The ECS ESST can also save the file to a user-selectable directory for later viewing by EOSView or other viewers. EOSView also supports viewing FTP browse on the user's workstation.

4.2.4.1 ODL Normalization Form for Browse Request

The ODL Normalization Form for FTP/integrated browse requests defined in this section pertains, in its entirety, to the following interfaces:

- (Release A only) ECS RASOT-to-EOSDIS V0 IMS Server
- (Release B only) ECS ESST (via the V0 Gateway)-to-EOSDIS V0 IMS Server

BROWSE_REQUEST group ::==
    MESSAGE_ID
    [AUTHENTICATOR]
    [ECS_AUTHENTICATOR]
    DATA_CENTER_ID
    USER_AFFILIATION group
    BROWSE_TYPE
    BROWSE_GRANULES group
    CONTACT_ADDRESS group
    MONITOR group
    VERSION group

BROWSE_GRANULES ::==
    DATASET_ID
    GRANULE_ID

CONTACT_ADDRESS group ::==
    TITLE
    LAST_NAME
    FIRST_NAME

[MIDDLE_INITIAL]
ORGANIZATION
ADDRESS
CITY
[STATE]
[ZIP]
COUNTRY
PHONE
[FAX]
EMAIL

MONITOR group ::==
TX_CLIENT
[RX_SERVER]
[TX_SERVER]
[RX_CLIENT]

VERSION group ::==
PROTOCOL_VERSION
SENDER_VERSION
[IMS_STAFF]

USER_AFFILIATION group ::==
CATEGORY
TYPE

4.2.4.2 ODL Normalization Form for FTP Browse Results

The ODL Normalization Form for the FTP browse results message defined in this section pertains, in its entirety, to the following interfaces:

- (Release A only) EOSDIS V0 IMS Server-to-ECS RASOT
- (Release B only) EOSDIS V0 IMS Server (via the V0 Gateway)-to-ECS ESST

FTP_BROWSE_RESULT group ::==
MESSAGE_ID
DATA_CENTER_ID
STATUS_CODE
[STATUS_CODE_COMMENT]
TOTAL_FILE_SIZE
DAAC_CONTACT_ADDRESS group
MONITOR group
VERSION group

DAAC_CONTACT_ADDRESS group ::==
CONTACT_NAME

ORGANIZATION
ADDRESS
CITY
STATE
ZIP
COUNTRY
PHONE
FAX
EMAIL

MONITOR group ::==
TX_CLIENT
RX_SERVER
TX_SERVER
[RX_CLIENT]

VERSION group ::==
PROTOCOL_VERSION
SENDER_VERSION
[IMS_STAFF]

4.2.4.3 ODL Normalization Form for Integrated Browse Results

The ODL Normalization Form for the integrated browse results message defined in this section pertains, in its entirety, to the following interfaces:

- (Release A only) EOSDIS V0 IMS Server-to-ECS RASOT
- (Release B only) EOSDIS V0 IMS Server (via the V0 Gateway)-to-ECS ESST

INTEGRATED_BROWSE_RESULT ::==
MESSAGE_ID
DATA_CENTER_ID
STATUS_CODE
[STATUS_CODE_COMMENT]
IMAGE group
MONITOR group

IMAGE group ::==
DATASET_ID
GRANULE_ID
IMAGE_ID
IMAGE_SIZE

MONITOR group ::==
TX_CLIENT
RX_SERVER

TX_SERVER
[RX_CLIENT]

The INTEGRATED_BROWSE_RESULT message is followed by the browse file itself transferred as a binary stream of IMAGE_SIZE bytes.

4.2.5 Product Request/Result

(Release A only) The Product Request allows the user to order V0 data products through the ECS RASOT. After the user has successfully searched, located, and viewed the inventory data for the datasets and selected the granules desired, and (possibly) after the user has viewed certain "representative" browse images, the user may (but is not required to) submit a product request. [It is important to note that with the RASOT there is no way for the user to order data without first doing an inventory search and viewing the inventory data---this option exists only for browse.] The Product Request is sent from the ECS RASOT to the EOSDIS V0 IMS Server. The Product Result is sent from the EOSDIS V0 IMS Server to the ECS RASOT.

(Release B only) The Product Request allows the user to order V0 data products through the ECS ESST. After the user has successfully searched, located, and viewed the inventory data for the datasets and selected the granules desired, and (possibly) after the user has viewed certain "representative" images, the user may (but is not required to) submit a product request. The Product Request is sent from applicable workbench tools such as ECS ESST (via the V0 Gateway) to the EOSDIS V0 IMS Server. The Product Result is sent from the EOSDIS V0 IMS Server (via the V0 Gateway) to applicable workbench tools such as ECS ESST.

The Product Result provides a confirmation of the archive's receipt of the Product Request and provides contact information for further inquiries. The actual product is distributed by the V0 System via hard media or to an appropriate FTP site.

4.2.5.1 ODL Normalization Form for Product Request

The ODL Normalization Form for product requests defined in this section pertains, in its entirety, to the following interfaces:

- (Release A only) ECS RASOT-to-EOSDIS V0 IMS Server
- (Release B only) ECS ESST (via the V0 Gateway)-to-EOSDIS V0 IMS Server

PRODUCT_REQUEST group ::==
MESSAGE_ID
REQUEST_ID
DATA_CENTER_ID
[AUTHENTICATOR]
[ECS_AUTHENTICATOR]
[INITIAL_USER_KEY]

USER_AFFILIATION group
CONTACT_ADDRESS group
[SHIPPING_ADDRESS] group
[BILLING_ADDRESS] group
(LINE_ITEM group)+ repeated group
MONITOR group
VERSION group

USER_AFFILIATION group ::==
CATEGORY
TYPE

CONTACT_ADDRESS group ::==
[TITLE]
LAST_NAME
FIRST_NAME
[MIDDLE_INITIAL]
[ORGANIZATION]
[ADDRESS]
CITY
[STATE]
[ZIP]
COUNTRY
PHONE
[FAX]
EMAIL (for FTP Browse and Prod. Req.)

SHIPPING_ADDRESS group ::== Optional group
[TITLE]
LAST_NAME
FIRST_NAME
[MIDDLE_INITIAL]
[ORGANIZATION]
[ADDRESS]
CITY
[STATE]
[ZIP]
COUNTRY
PHONE
[FAX]
EMAIL

BILLING_ADDRESS group ::== Optional group
[TITLE]
LAST_NAME
FIRST_NAME

[MIDDLE_INITIAL]
[ORGANIZATION]
[ADDRESS]
CITY
[STATE]
[ZIP]
COUNTRY
PHONE
[FAX]
EMAIL

LINE_ITEM group ::==
DATASET_ID
PACKAGE_ID
PROCESSING_OPTION
MEDIA_TYPE
MEDIA_FORMAT
[ADDITIONAL_INFO]
[BILLING_ID]
[EST_COST]

MONITOR group ::==
TX_CLIENT
[RX_SERVER]
[TX_SERVER]
[RX_CLIENT]

VERSION group ::==
PROTOCOL_VERSION
SENDER_VERSION
[IMS_STAFF]

4.2.5.2 ODL Normalization Form for Product Results

The ODL Normalization Form for the product results message defined in this section pertains, in its entirety, to the following interfaces:

- (Release A only) EOSDIS V0 IMS Server-to-ECS RASOT
- (Release B only) EOSDIS V0 IMS Server (via the V0 Gateway)-to-ECS ESST

PRODUCT_RESULT group ::==
MESSAGE_ID
DATA_CENTER_ID
STATUS_CODE
[STATUS_CODE_COMMENT]

(DAAC_CONTACT_ADDRESS group)+ (repeatable mostly to support "DAACs" that are consortia of multiple archives in the international community)
MONITOR group

DAAC_CONTACT_ADDRESS group ::==
CONTACT_NAME
ORGANIZATION
ADDRESS
CITY
STATE
ZIP
COUNTRY
PHONE
FAX
EMAIL

MONITOR group ::==
TX_CLIENT
[RX_SERVER]
[TX_SERVER]
[RX_CLIENT]

4.2.6 Quit

(Release A only) During any given session, problems may necessitate premature termination of the process. In such cases, a bi-directional quit message is transmitted between the EOSDIS V0 IMS Server and the ECS RASOT, as appropriate. Specifically, the ECS RASOT sends a quit message to the EOSDIS V0 IMS Server if the user presses the "abort" button on the screen. On the other hand, the quit message is sent by the EOSDIS V0 IMS Server to the ECS RASOT if an error condition terminates the response. Quit messages are also used to synchronize the ECS RASOT with the EOSDIS V0 IMS Server following the last chunk in an inventory result---the EOSDIS V0 IMS Server sends a QUIT with a STATUS_CODE of 1 to the ECS RASOT and the ECS RASOT sends a similar QUIT back to the EOSDIS V0 IMS Server.

(Release B only) During any given session, problems may necessitate premature termination of the process. In such cases, a bi-directional quit message is transmitted, via the V0 Gateway, between the EOSDIS V0 IMS Server and the ECS ESST, as appropriate. Specifically, the ECS ESST sends a quit message, via the V0 Gateway, to the EOSDIS V0 IMS Server if the user presses the "abort" button on the screen. On the other hand, the quit message is sent by the EOSDIS V0 IMS Server, via the V0 Gateway, to the ECS ESST if an error condition terminates the response. Quit messages are also used to synchronize the ECS ESST with the EOSDIS V0 IMS Server following the last chunk in an inventory result. The EOSDIS V0 IMS Server sends a QUIT with a STATUS_CODE of 1, via the V0 Gateway, to the ECS ESST and the ECS ESST sends a similar QUIT back to the EOSDIS V0 IMS Server via the V0 Gateway.

4.2.6.1 ODL Normalization Form for Quit

The ODL Normalization Form for the bi-directional quit message defined in this section pertains, in its entirety, to the following interfaces:

- (Release A only) ECS RASOT-to-EOSDIS V0 IMS Server
- (Release B only) ECS ESST (via the V0 Gateway)-to-EOSDIS V0 IMS Server

QUIT group ::==
    MESSAGE_ID
    [DATA_CENTER_ID]
    STATUS_CODE
    [STATUS_CODE_COMMENT]

    MONITOR group
    VERSION group

MONITOR group ::==
    TX_CLIENT
    [RX_SERVER]
    [TX_SERVER]
    [RX_CLIENT]

VERSION group ::==
    PROTOCOL_VERSION
    SENDER_VERSION
    [IMS_STAFF]

4.2.7 Pong (Release A only)

At Release A, the pong message is used to test for the existence, proper operation, availability and responsiveness of the EOSDIS V0 IMS Server. The pong message consists of a small amount of data and is time-tagged. This bi-directional message is invoked at regular intervals by the ECS RASOT, and passed from the RASOT to the "Ponger" Program for transmission to the EOSDIS V0 IMS Server. (Note: the "Ponger" program is an executable that is separate from, but spawned by, the ECS RASOT.) The pong message is then echoed back by the EOSDIS V0 IMS Server. Upon return the turn-around time is determined. The pong message is used to establish which DAAC is "time-closest," and this information is used, in turn, for load balancing for outgoing statistics messages.

4.2.7.1 ODL Normalization Form for Pong (Release A only)

The ODL Normalization Form for the pong message is presented below:

PONG group ::==
    MESSAGE_ID
    TX_CLIENT
    [TX_SERVER]

MONITOR group
VERSION group ::==

MONITOR group ::==
[TX_CLIENT]
[RX_SERVER]
[TX_SERVER]
[RX_CLIENT]

VERSION group ::==
PROTOCOL_VERSION
SENDER_VERSION
[IMS_STAFF]

4.2.8 Statistics (Release A only)

At Release A, the suite of statistics messages consists of three messages which are used to collect statistics from the ECS RASOT to be sent to the EOSDIS V0 IMS Server for archiving and forwarding to the statistics group---these messages include the session start, session end, and product request report messages. Typical statistics collected include the length of a user session and the number of searches performed by the ECS RASOT. The statistics messages are invoked by the ECS RASOT, and passed to the "Ponger" executable (see section 4.2.7) for transmission to the EOSDIS V0 IMS Server. If the Ponger cannot be run, these messages are transferred using e-mail.

- **Session Start -** The session start message is used together with the session end message (see below) to calculate the number of user sessions which ended abnormally. Barring a start-up failure, there will always be one session start message for each session. Typical information contained in the session start message includes: client operating system, IP address, whether the user is an IMS staffer, client version, etc.
- **Session End -** The session end message is used together with the session start message (see above) to calculate the number of user sessions which ended abnormally. The session end message is only generated when the client exits normally. Typical information contained in the session end message includes: number of searches, number of browse images, average responsiveness, etc.
- **Product Request Report -** The product request report message is used to determine the number of items ordered during a product request transaction. One product request report message is transmitted each time the user submits an order, e.g., presses the "Submit Order" button on the ECS RASOT's graphical user interface (GUI). Typical information contained in the product request report includes: number of orders, what was ordered, what media, etc.

4.2.8.1 ODL Normalization Form for Statistics (Release A only)

The ODL Normalization Form for the session start, session end, and product request report messages is presented below:

STATS_REPORT group ::==
MESSAGE_ID
STATS_REPORT_LENGTH
STATS_REPORT_STRING
MONITOR group
VERSION group ::==

MONITOR group ::==
SESSION_ID
[TX_CLIENT]
[RX_SERVER]
[TX_SERVER]
[RX_CLIENT]

VERSION group ::==
PROTOCOL_VERSION
SENDER_VERSION
[IMS_STAFF]

4.3 Data Flows Between EOSDIS V0 IMS Client (via V0 Gateway) and ECS Servers

The data flows between the EOSDIS V0 IMS Client and ECS are discussed in this section and are depicted in Figure 4-5 [it should be noted that data flows between the ECS RASOT and the V0 IMS Servers (at Release A) and between the ECS ESST (via the V0 Gateway) and the V0 IMS Servers (at Release B) are discussed in section 4.2]. Specifically, the following data flows are depicted:

- Between EOSDIS V0 IMS Client and ECS Science Data Server (via the V0 Gateway):
    - Directory Search Request
    - Directory Search Results
    - Inventory Search Request
    - Inventory Search Results
    - Acknowledge
    - Browse Request
    - Browse Results
    - Product Request
    - Product Results
    - Quit

- Between EOSDIS V0 IMS Client and the V0 Gateway:
    - Statistics
    - Pong
- Between EOSDIS V0 IMS Client and ECS Document Data Server:
    - Guide Search Request
    - Guide Search Results

Between the EOSDIS V0 IMS CLIENT and the ECS Document Data Servers, WAIS formatted queries over HTTP protocol are used for guide request/results. With the exception of the guide search request/results messages, all messages use Object Description Language (ODL). (For a description of ODL refer to the User's Guide for the Object Description Language Processing Software Library, Release 2.1 - Draft.) All of these messages are handled by the IMS Kernel (IK) layer [Note: The EOSDIS V0 IMS CLIENT and the V0 Gateway contain several software modules, at the communications (lowest) layer, which serve as library routines and are, collectively, referred to as the IK layer.] Pong and statistics messages are invoked by the EOSDIS V0 IMS Client and transmitted by the "Ponger" program (an executable that is separate from, but spawned by, the EOSDIS V0 IMS Client) to the V0 Gateway.

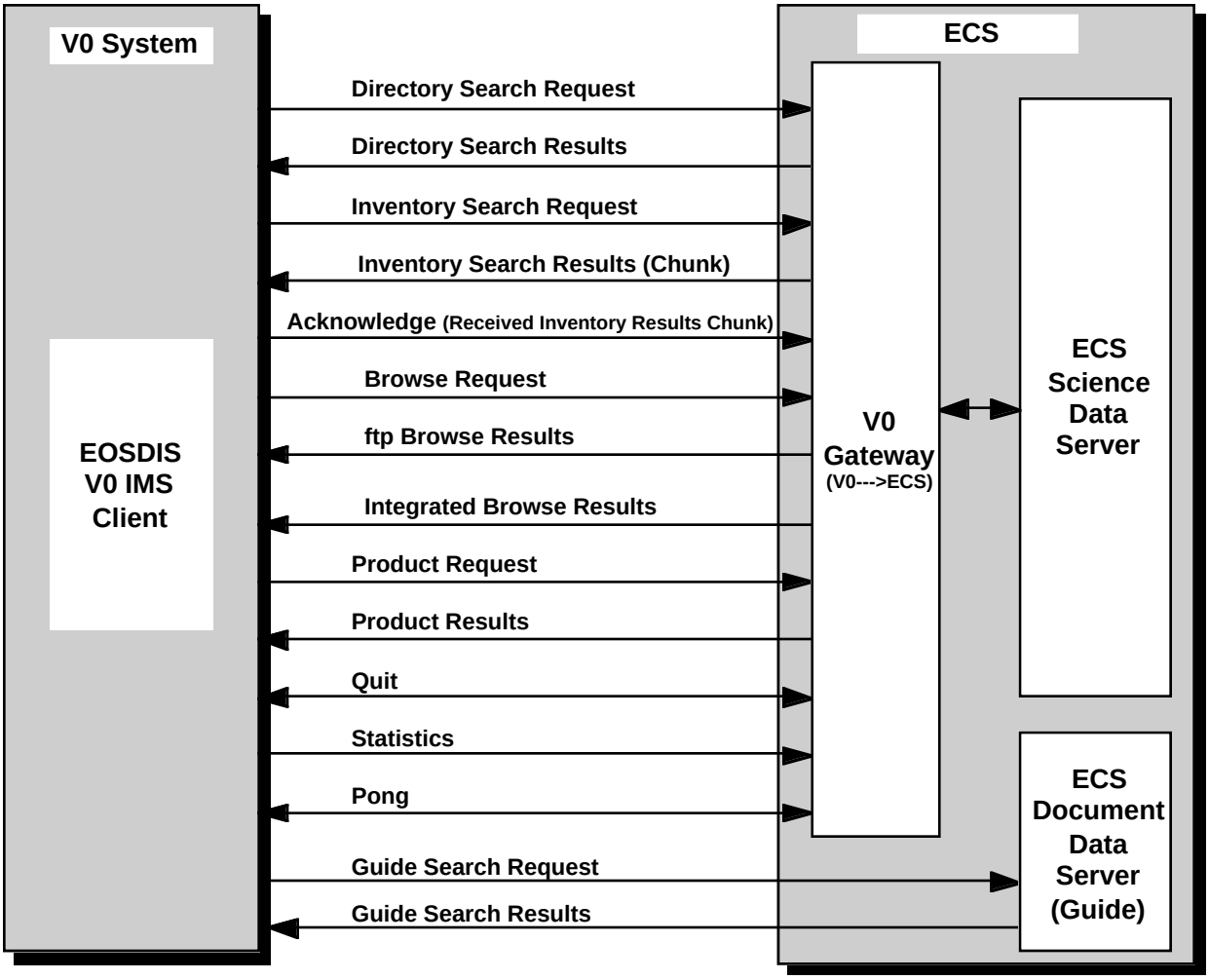


Figure 4-5. Interfaces Between EOSDIS V0 IMS Client (via V0 Gateway) and ECS Servers

4.3.1 Directory Search Request/Results

The purpose of the directory search is to aid the user in making an initial determination of the potential usefulness of various data sets pertinent to some application by searching through descriptions of metadata or data set catalogues which contain high-level information. The directory search provides information on the location of metadata or data set catalogues. The search criteria, specified by the user, are based on the following typical searchable attributes: source, sensor, geophysical parameter, dataset name, data center id, geographical coordinates (area), temporal intervals, etc. A V0 user, requesting ECS services, submits the directory search request via the EOSDIS V0 IMS Client. The EOSDIS V0 IMS Client sends the request, via the V0 Gateway, to the ECS Science Data Server. The ECS Science Data Server returns MD_ENTRY_ID keywords, via the V0 Gateway, to the EOSDIS V0 IMS Client. The EOSDIS V0 IMS Client accesses the GCMD using these MD_ENTRY_ID keywords. The GCMD then returns directory information to the EOSDIS V0 IMS Client. The Directory Search Request and Directory Search Results messages are implemented using ODL---their ODL Normalization Forms are defined in the immediately-following sections. [A discussion of the ODL standardized conventions is provided as reference in Section 4.1. Detailed definitions of the message keywords (e.g., MESSAGE_ID) are provided in Section 5.]

4.3.1.1 ODL Normalization Form for Directory Search Request

The ODL Normalization Form for the EOSDIS V0 IMS Client (via the V0 Gateway)-to-ECS Science Data Server Directory Search Request message is equivalent to that defined in Section 4.2.1.1.

4.3.1.2 ODL Normalization Form for Directory Search Results

The ODL Normalization Form for the ECS Science Data Server (via the V0 Gateway)-to-EOSDIS V0 IMS Client Directory Search Results message is equivalent to that defined in Section 4.2.1.2.

4.3.2 Guide Search Request/Results

The purpose of the guide search is to locate and retrieve guide documents containing detailed information about datasets, based on user-specified keywords or freetext strings; it also allows a user to locate datasets if (s)he decides not to go through a directory or inventory search first. In addition, guide serves to aid a user in understanding the metadata in the search construction process. The search criteria, specified by the user, are based on the following typical searchable attributes: source, sensor, geophysical parameter, dataset name, data center id, geographical coordinates (area), temporal intervals, etc. A V0 user can look through the ECS guide documents, follow any hypertext link and display any related documents. The guide documents give detailed information on campaign, source, sensor, data_center, and dataset_id.

4.3.2.1 Guide Search Request

The EOSDIS V0 IMS Client sends search requests for documents as WAIS formatted queries over HTTP protocol. The valids processing in the EOSDIS V0 IMS Client determines which guide servers to send the guide query to, and then creates a separate search message for each destination server, based on the information in the archive.odl file (an example is depicted in Figure 4-6). The archive.odl file configured with each EOSDIS V0 IMS Client identifies the ECS Document Data Servers as HTTP servers---specifically, Figure 4-6 depicts an example of the archive.odl file documenting server address and HTTP protocol for connecting to an ECS Document Data Server (bold emphasis added). The parameters and values in the guide search queries are of the same domain as the search and order parameters and values used for the Directory and Inventory search functions in the EOSDIS V0 IMS Client, but may be different from the valids used internally at the ECS DAACs. Each ECS Document Data Server will perform a valids mapping using the function provided by the V0 Gateway. The HTTP servers at the ECS DAACs receive the guide queries, perform the mapping, and process the search using the Science Data Server Data Base Management System (DBMS) interfaces.

```
/* $Id: archive.odl,v 4.3.4.1 1995/08/02 17:08:26 ims Exp $ */

/* OPERATIONAL/STABLE/DEMO archive information */

GROUP              = DATA_CENTER_INFO

  GROUP       = DATA_CENTER
     DATA_CENTER_ID              = "[GSFC_TBS]"
     DATA_CENTER_NAME    = "GODDARD SPACE FLIGHT CENTER"
     INTERNET            = "ddsrv.hitc.com"
     PORT               = "8080"
    GUIDE_SRV_ADDR      = "http://ddsrv.hitc.com:8080/cgi-bin/ddsrv"
  END_GROUP   = DATA_CENTER
                                        o
                                        o
```

Figure 4-6. Example of Archive.odl File Documenting Server Address and HTTP Protocol for Connecting to an ECS Document Data Server

(bold emphasis added) Note: Naming conventions for the DAACs are noted as "[DAAC name]_TBS", to distinguish them from the Version 0 DAACs in the archive.odl file.

4.3.2.2 Guide Search Results

Each of the ECS Document Data Servers creates a results "hit" list formatted with HTML, and sends the HTML page over the HTTP connection, independently of the other ECS Document Data Servers. The EOSDIS V0 IMS Client does not integrate the results of the multiple ECS Document Data Servers into a merged list. Each document "hit" is displayed as a hyperlink in the guide results window in the EOSDIS V0 IMS Client. When the user selects a guide document for viewing, the request is sent as an HTTP message using the HTTP protocol. This HTTP message contains the path name of document and the server address which stores the document. The document is returned to the EOSDIS V0 IMS Client, over the HTTP connection.

4.3.3 Inventory Search Request/Results and Acknowledgment

The purpose of the inventory search is to aid a user in searching through the available inventory, locating and retrieving metadata about specific granules of the product(s) of interest, and determining whether any granules should be ordered; and also to allow a user to find datasets if the user chooses not to use a directory or guide search first. The search criteria, specified by the user, are based on the following searchable attributes: source, sensor, geophysical parameter, dataset name, data center id, geographical coordinates (area), temporal intervals, etc. A V0 user, requesting ECS services, submits the inventory search request via the EOSDIS V0 IMS Client. The EOSDIS V0 IMS Client sends (via the V0 Gateway) the ECS Science Data Server inventory search criteria based on characteristics of the data. The ECS Science Data Server retrieves the requested granules' metadata, and sends these items (via the V0 Gateway) back to the EOSDIS V0 IMS Client in chunks of various sizes. The EOSDIS V0 IMS Client returns a separate acknowledge message to the ECS Science Data Server (via the V0 Gateway) upon receiving each chunk (the "chunking protocol" is described in section 4.2.3).

The Inventory Search Request and Inventory Search Results messages are implemented using ODL---their ODL Normalization Forms are defined in the immediately-following sections. [A discussion of the ODL standardized conventions is provided as reference in Section 4.1. Detailed definitions of the message keywords (e.g., MESSAGE_ID) are provided in Section 5.]

In order to accommodate two-way mapping of terminology between ECS and the V0 System, the V0 Gateway (a part of ECS) maintains a Sybase database containing the terminology mapping information. The V0 Gateway database is built by a Gateway Administrator using V0 System search parameters, ECS schema and metadata. Specifically, upon receiving a request from the EOSDIS V0 IMS Client, the V0 Gateway performs a V0-ECS mapping table look-up within the V0 Gateway database, converting the V0 request into ECS's terminology in order to accommodate the ECS Science Data Server. Similarly, results returned from the ECS Science Data Server to the V0 Gateway are converted, via the V0-ECS mapping service, to V0 terminology prior to returning these results to the EOSDIS V0 IMS Client. The V0 Gateway-to-Sybase and V0 Gateway-to-ECS Science Data Server interfaces are internal to ECS, and are completely documented in CDRL #305-CD-023-002, Release B SDPS Data Management Subsystem Design Specification for the ECS Project.

4.3.3.1 ODL Normalization Form for Inventory Search Request

The ODL Normalization Form for the EOSDIS V0 IMS Client (via the V0 Gateway)-to-ECS Science Data Server Inventory Search Request message is equivalent to that defined in Section 4.2.3.1.

4.3.3.2 ODL Normalization Form for Inventory Search Results

The ODL Normalization Form for the ECS Science Data Server (via the V0 Gateway)-to-EOSDIS V0 IMS Client Inventory Search Results message is equivalent to that defined in Section 4.2.3.2.

4.3.3.3 ODL Normalization Form for Acknowledge

The ODL Normalization Form for the EOSDIS V0 IMS Client (via the V0 Gateway)-to-ECS Science Data Server Acknowledge message is equivalent to that defined in Section 4.2.3.3.

4.3.4 Browse Request/Results

The purpose of the Browse service is to allow the user to request and receive "representative" images for FTP-copying or viewing and for analysis prior to deciding on specific full-resolution products to order. Two browse service modes are available to a user, including FTP browse and integrated browse.

The FTP Browse service allows the user to FTP-copy the browse product onto the user's system. After the ECS Science Data Server receives an FTP browse request (BROWSE_TYPE = FTP_Only), it transmits the FTP Browse Results to the EOSDIS V0 IMS Client and copies the browse product onto the DAAC FTP site. The user is notified, via e-mail that the browse product is ready to FTP. This notification provides the access information for the FTP including the following: (1) FTP account, (2) IP address of the FTP account, (3) directory and file name for each of the requested browse products, (4) pick up expiration date of staged files. The FTP browse product remains on the FTP site for an operations-tunable time interval, giving the user time to copy it onto the user's system.

The Integrated Browse service allows the user to view the browse product through the EOSDIS V0 IMS Client. In response to an integrated browse request (BROWSE_TYPE = Y) sent by the EOSDIS V0 IMS Client, via the V0 Gateway, to the ECS Science Data Server, the ECS Science Data Server sends back to the EOSDIS V0 IMS Client (via the V0 Gateway) the integrated browse results message, followed by the browse image, which is then displayed to the user.

All ECS Browse images (FTP and integrated) are provided in HDF-EOS (an extension of HDF, Version 4.0, from the NCSA) which is supported by the ECS Science Data Server.

The Browse Request/Results messages are implemented using ODL---their ODL Normalization Forms are defined in the immediately-following sections. [A discussion of the ODL standardized conventions is provided as a reference in Section 4.1. Detailed definitions of the message keywords (e.g., MESSAGE_ID) are provided in Section 5.]

The EOSDIS V0 IMS Client can display the image layers of ECS browse data files written in HDF-EOS format. This will help the V0 user to visualize ECS browse images during the selection of data and to verify that the data received is the data desired. It is important to point out that the EOSDIS V0 IMS Client is not capable of displaying text, table or movie loop documents. The EOSDIS V0 IMS Client can also save a browse file in a user-selectable directory for viewing with other viewers such as EOSView. EOSView also supports viewing FTP browse on the user's workstation.

4.3.4.1 ODL Normalization Form for Browse Request

The ODL Normalization Form for the EOSDIS V0 IMS Client (via the V0 Gateway)-to-ECS Science Data Server Browse Request message is equivalent to that defined in Section 4.2.4.1.

4.3.4.2 ODL Normalization Form for FTP Browse Results

The ODL Normalization Form for the ECS Science Data Server (via the V0 Gateway)-to-EOSDIS V0 IMS Client FTP Browse Results message is equivalent to that defined in Section 4.2.4.2.

4.3.4.3 ODL Normalization Form for Integrated Browse Results

The ODL Normalization Form for the ECS Science Data Server (via the V0 Gateway)-to-EOSDIS V0 IMS Client Integrated Browse Results message is equivalent to that defined in Section 4.2.4.3.

4.3.5 Product Request/Result

The Product Request allows the user to order ECS data products through the EOSDIS V0 IMS Client. After the user has successfully searched, located, and viewed the inventory data for the datasets and selected the granules desired; and (possibly) after the user has viewed certain "representative" browse images, the user may (but is not required to) submit a product request. The Product Request is sent from the EOSDIS V0 IMS Client, via the V0 Gateway, to the ECS Science Data Server. [It is important to note that with the EOSDIS V0 IMS Client there is no way for the user to order data without first doing an inventory search and viewing the inventory data---this option exists only for browse.] The Product Result is sent from the ECS Science Data Server, via the V0 Gateway, to the EOSDIS V0 IMS Client. The Product Result provides a confirmation of the ECS Science Data Server's receipt of the Product Request and provides contact information for further inquiries. The actual product is distributed by ECS via hard media or to an appropriate FTP site.

4.3.5.1 ODL Normalization Form for Product Request

The ODL Normalization Form for the EOSDIS V0 IMS Client (via the V0 Gateway)-to-ECS Science Data Server are equivalent to those in Section 4.2.5.1.

4.3.5.2 ODL Normalization Form for Product Result

The ODL Normalization Form for the ECS Science Data Server (via the V0 Gateway)-to-EOSDIS V0 IMS Client are equivalent to those in Section 4.2.5.2.

4.3.6 Quit

During any given session, problems may necessitate premature termination of the process. In such cases, a bi-directional quit message is transmitted between the ECS Science Data Server (via the V0 Gateway) and the EOSDIS V0 IMS Client, as appropriate. Specifically, the EOSDIS V0 IMS Client sends, via the V0 Gateway, a quit message to the ECS Science Data Server if the user presses the "abort" button on the screen. On the other hand, the quit message is sent by the ECS Science Data Server, via the V0 Gateway, to the EOSDIS V0 IMS Client if an error condition terminates the response. Quit messages are also used to synchronize the EOSDIS V0 IMS Client with the ECS Science Data Server following the last chunk in an inventory result--- the ECS Science Data Server sends a QUIT with a STATUS_CODE of 1, via the V0 Gateway, to the EOSDIS V0 IMS Client and the EOSDIS V0 IMS Client sends a similar QUIT back to the ECS Science Data Server, via the V0 Gateway.

4.3.6.1 ODL Normalization Form for Quit

The ODL Normalization Form for the ECS Science Data Server (via the V0 Gateway)-to-EOSDIS V0 IMS Client Quit message is equivalent to that defined in Section 4.2.6.1.

4.3.7 Pong

The pong message is used to test for the existence, proper operation, availability and responsiveness of the V0 Gateway. The pong message consists of a small amount of data and is time-tagged. This bi-directional message is invoked at regular intervals by the EOSDIS V0 IMS Client, and passed to the "Ponger" Program for transmission to the V0 Gateway. (Note: the "Ponger" program is an executable that is separate from, but spawned by, the EOSDIS V0 IMS Client.) The pong message is then echoed back via the V0 Gateway. Upon return the turn-around time is determined. The pong message is used to establish which DAAC is "time-closest," and this information is used, in turn, for load balancing for outgoing statistics messages.

4.3.7.1 ODL Normalization Form for Pong

The ODL Normalization Form for the pong message is equivalent to that defined in Section 4.2.7.1.

4.3.8 Statistics

The suite of statistics messages consists of three messages which are used to collect statistics from the EOSDIS V0 IMS Client to be sent to the V0 Gateway for forwarding to the statistics group---these messages include the session start, session end, and product request report messages. Typical statistics collected include the length of a user session and the number of

searches performed by the EOSDIS V0 IMS Client. The statistics messages are invoked by the EOSDIS V0 IMS Client and passed to the "Ponger" executable (see section 4.3.7) for transmission to the V0 Gateway. If the Ponger cannot be run, these messages are transferred using e-mail.

- **Session Start -** The session start message is used together with the session end message (see below) to calculate the number of user sessions which ended abnormally. Barring a start-up failure, there will always be one session start message for each session. Typical information contained in the session start message includes: client operating system, IP address, whether the user is an IMS staffer, client version, etc.
- **Session End -** The session end message is used together with the session start message (see above) to calculate the number of user sessions which ended abnormally. The session end message is only generated when the client exits normally. Typical information contained in the session end message includes: number of searches, number of browse images, average responsiveness, etc.
- **Product Request Report -** The product request report message is used to determine the number of items ordered during a product request transaction. One product request report message is transmitted each time the user submits an order, e.g., presses the "Submit Order" button on the EOSDIS V0 IMS Client graphical user interface (GUI). Typical information contained in the product request report includes: number of orders, what was ordered, what media, etc.

4.3.8.1 ODL Normalization Form for Statistics

The ODL Normalization Form for the session start, session end, and product request report messages is equivalent to that defined in Section 4.2.8.1.

5. ODL Message Keywords (Objects)

This section identifies and defines each of the ODL Message keywords corresponding to the ODL descriptions provided in section 4 of this document. Each keyword is defined, as applicable, in terms of synopsis (short English-Language description of the keyword), parent groups, children, ODL type [e.g., integer, real, date, string, aggregate (i.e., the keyword object contains children), symbol, sequence string (i.e., 0 or more strings entered on separate lines), and character string.], maximum (value) length, and possible values. If no possible values are specified, then any possible value for the stated ODL type is legal. For example, an ACCOUNT_NUMBER may be any string up to 80 characters. The ODL keywords described in this section are derived from the "Messages and Development Data Dictionary - v0 and Release A Message Passing Protocol Specification," 9/95.

Keyword: ACCOUNT_NUMBER
Synopsis: Account identifier provided by a DAAC.
Parent Group(s): VALID_ACCOUNT

ODL Type: String
Maximum Length: 80

Keyword: ADDITIONAL_INFO
Synopsis: User supplied information about user's order, and is applied to each package ordered.
Parent Group(s): LINE_ITEM

ODL Type: String
Maximum Length: 80

Keyword: ADDRESS
Synopsis: Address information can be entered using three lines.
Parent Group(s): BILLING_ADDRESS, CONTACT_ADDRESS, SHIPPING_ADDRESS, DAAC_CONTACT_ADDRESS
ODL Type: Sequence String
Field length: 32 x 3 (96)

Keyword: APPROX_COST
Synopsis: Estimated cost for the selected data package.
Parent Group(s): MEDIA
ODL Type: Real
Maximum Length: 16
Possible value(s): 0.0 to 9999999999999.99

Keyword: AUTHENTICATON_KEY
Synopsis: Password provided by user for accessing restricted data (at Data Centers).

ODL Type: String
Maximum Length: 16

Keyword: AUTHENTICATOR
Synopsis: Encrypted value from authentication key, last name, first name. Passed with every request (if authentication key is not blank).
Parent Group(s): BROWSE_REQUEST, PRODUCT_REQUEST, INVENTORY_SEARCH, DIRECTORY_SEARCH
ODL Type: String
Maximum Length: 16

Keyword: BALANCE
Synopsis: Dollar amount remaining for a particular account.
Parent Group(s): VALID_ACCOUNTS

ODL Type: Real
Maximum Length: 16

Keyword: BILLING_ADDRESS
Synopsis: Billing address for data order.
Parent Group(s): PRODUCT_REQUEST
Child Group(s): CITY, EMAIL, FAX, FIRST_NAME, MIDDLE_INITIAL, LAST_NAME, PHONE, STATE, COUNTRY, ZIP, TITLE, ORGANIZATION
ODL Type: Aggregate

Keyword: BILLING_ID
Synopsis: Account number that the user enters or selects.
Parent Group(s): LINE_ITEM
ODL Type: String
Maximum Length: 80

Keyword: BROWSE_GRANULES
Synopsis: granule(s) request
Parent Group(s): BROWSE_REQUEST
Child Group(s): DATASET_ID, GRANULE_ID (sequence)
ODL Type: Aggregate
Note: pass all GRANULE_IDs for FTP Browse requests in one BROWSE_REQUEST group, putting all GRANULE_IDs of a dataset in one sequence within the BROWSE_GRANULES group; [This is for multiple FTP browse. Currently the client does not do this, though it is preferable for the servers to be ready to handle it.] It sends one request group for each FTP browse and also BROWSE-REQUEST group for each browse. Current Integrated Browse requests should be sent one at a time, each in a separate BROWSE_REQUEST group.

Keyword: BROWSE_ONLY
Synopsis: Only granules with associated browse images should be returned from the INVENTORY_SEARCH.
Parent Group(s): INVENTORY_SEARCH
ODL Type: Symbol
Maximum Length: 1
Possible value(s): Y

Keyword: BROWSE_PRODUCT_DESCRIPTION
Synopsis: Dataset specific browse product (image) description
Parent Group(s): DATASET

ODL Type: Sequence String
Maximum Length: 80

Keyword: BROWSE_REQUEST
Synopsis: Provide information for obtaining browse image
Child Group(s): BROWSE_TYPE, MESSAGE_ID, MONITOR group, CONTACT_ADDRESS group, BROWSE_GRANULES group, AUTHENTICATOR, DATA_CENTER_ID, VERSION group
ODL Type: Aggregate

Keyword: BROWSE_TYPE
Synopsis: Type of delivery for browse image
Parent Group(s): BROWSE_REQUEST, GRANULE
ODL Type: Symbol
Maximum Length: 8
Possible value(s): Y | N | FTP_Only
Notes:
If Y is in a request, then = 'send integrated browse'.
If Y is in a granule, then = 'available in integrated and FTP browse'.
If N is in a granule, then = 'not available'.
If FTP is in request, then = 'send FTP browse only'.
If FTP is in granule, then = 'available only as FTP'.

Keyword: CAMPAIGN
Synopsis: Name of campaign/project that gathered data.
Parent Group(s): [DIRECTORY_SEARCH], [DATASET], [GRANULE], [INVENTORY_SEARCH]
ODL Type: Sequence String
Maximum Length: 80

Keyword: CATEGORY
Synopsis: Affiliation category for a user
Parent Group(s): USER_AFFILIATION
ODL Type: String
Maximum Length: 7

Possible value(s): USA, NOT USA

Keyword: CENTROID_LAT
Synopsis: Used for part of center point coordinate in the case where a granule is described as a polygon.
Parent Group(s): POLYGON_LOC group for INVENTORY_RESULTS
ODL Type: Real
Maximum Length: 8

Keyword: CENTROID_LON
Synopsis: Used for part of center point coordinate in the case where a granule is described as a polygon.
Parent Group(s): POLYGON_LOC group for INVENTORY_RESULTS
ODL Type: Real
Maximum Length: 8

Keyword: CITY
Synopsis: Name of the city of the associated address
Parent Group(s): BILLING_ADDRESS, CONTACT_ADDRESS, SHIPPING_ADDRESS, DAAC_CONTACT_ADDRESS
ODL Type: String
Maximum Length: 30
Possible value(s): any string

Keyword: COMMENT
Synopsis: Data Center provided information about corresponding granule or data set.
Parent Group(s): DATASET, GRANULE
ODL Type: Sequence String
Maximum Length: 60
Possible value(s): any string

Keyword: CONTACT_ADDRESS
Synopsis: The address portion of a user's contact information.
Parent Group(s): BROWSE_REQUEST, PRODUCT_REQUEST
Child Group(s): USER_AFFILATION, CITY, EMAIL, FAX, FIRST_NAME, MIDDLE_INITIAL, LAST_NAME, PHONE, STATE, COUNTRY, STREET, ZIP
ODL Type: Aggregate

Keyword: COUNTRY
Synopsis: The name for the country of the associated address
Parent Group(s): SHIPPING_ADDRESS, BILLING_ADDRESS, CONTACT_ADDRESS, DAAC_CONTACT_ADDRESS
ODL Type: String
Maximum Length: 30

Keyword: DAAC_CONTACT_ADDRESS
Synopsis: The Data Center's User Services Office contact information.
Parent Group(s): FTP_BROWSE_REQUEST group, PRODUCT_RESULT group

ODL Type: Aggregate

Keyword: DATA_CENTER_ID
Synopsis: Acronym form of the name of data center transmitting message.
Parent Group(s): DIRECTORY_RESULT, FTP_BROWSE_RESULT, INTEGRATED_BROWSE_RESULT, INVENTORY_RESULT, PRODUCT_RESULT, PRODUCT_REQUEST
ODL Type: Sequence String
Maximum Length: 10

Keyword: DATASET
Synopsis: Group to describe a dataset and associated granules from the result set
Parent Group(s): DIRECTORY_RESULT, INVENTORY_RESULT
Child group(s): COMMENT, DATASET_ID, GRANULE group, [MD_ENTRY_ID], NUMBER_OF_GRANULE_HITS, PARAMETER, [RESTRICTION], SENSOR_NAME, SOURCE_NAME, STATUS_CODE
ODL Type: Aggregate

Keyword: DATASET_ID
Synopsis: Name(s) of valid IMS data set(s)
Parent Group(s): BROWSE_REQUEST, DATASET, [DIRECTORY_SEARCH], IMAGE, [INVENTORY_SEARCH]
ODL Type: Sequence String
Maximum Length: 80

Keyword: DAY_NIGHT
Synopsis: Data gathered during "day" or "night"
Parent Group(s): [GRANULE], [DATASET]
ODL Type: Symbol
Maximum Length: 1
Possible value(s): D | N
Notes: DATASET unique and is under review.

Keyword: DIRECTORY_RESULT
Synopsis: Provides result of directory level query against data center.

Child Group(s): DATA_CENTER_ID, DATASET Group, MD_ENTRY_ID, MESSAGE_ID, MONITOR group, NUMBER_OF_DATASETS, STATUS_CODE
ODL Type: Aggregate
Notes:

1. In turn this information is used to query the GCMD.
2. The DATASET group in this case only requires the DATASET_ID and MD_ENTRY_ID.

Keyword: DIRECTORY_SEARCH
Synopsis: Provides data for directory level search of data center

Child Group(s): DATASET_ID, MESSAGE_ID, MONITOR group, RANGE_LOC group, [CAMPAIGN], [PARAMETER], [SENSOR_NAME], [SOURCE_NAME], [START_DATE], [STOP_DATE]
ODL Type: Aggregate

Keyword: EAST_LONGITUDE
Synopsis: Eastern most longitude for an area on the globe
Parent Group(s): RANGE_LOC
ODL Type: Real
Maximum Length: 9
Possible value(s): -180.0000 to +180.0000

Keyword: ECS_AUTHENTICATOR
Synopsis: Optional in every outgoing client message. Used for interfacing with ECS registration.
ODL Type: String
Maximum Length: 100

Keyword: EMAIL
Synopsis: Internet e-mail address for associated person
Parent Group(s): BILLING_ADDRESS, CONTACT_ADDRESS, SHIPPING_ADDRESS
ODL Type: String
Maximum Length: 128
Possible value(s): any string

Keyword: ERROR
Synopsis: Data Center provided freetext information about VALID_ACCOUNTS details. Provides multiple line of information.
Parent Group(s): VALID_ACCOUNTS

ODL Type: Sequence string
Maximum Length: 80

Keyword: EST_COST
Synopsis: Estimated cost of package
Parent Group(s): LINE_ITEM
ODL Type: Real
Maximum Length: 16
Possible value(s): 0.0 to 9999999999999.99

Keyword: FAX
Synopsis: FAX phone number for associated person
Parent Group(s): BILLING_ADDRESS, CONTACT_ADDRESS, SHIPPING_ADDRESS
ODL Type: String
Maximum Length: 22

Possible value(s): any string

Keyword: FIRST_NAME
Synopsis: The user's first name
Parent Group(s): BILLING_ADDRESS, CONTACT_ADDRESS, SHIPPING_ADDRESS
ODL Type: String
Maximum Length: 20
Possible value(s): any string

Keyword: FTP_BROWSE_RESULT
Synopsis: Provides results from a BROWSE_REQUEST
Child Group(s): CONTACT_ADDRESS group, DATA_CENTER_ID, MESSAGE_ID, MONITOR group, STATUS_CODE
ODL Type: Aggregate

Keyword: GCMD_SEARCH
Synopsis: Gaea internal ODL tree for GCMD search
Child Group(s): DATA_CENTER_ID, DATASET_ID, MD_ENTRY_ID, ORG_CENTER
ODL Type: Aggregate

Keyword: GLOBAL_GRANULE
Synopsis: Granule has global coverage
Parent Group(s): GRANULE
ODL Type: Symbol
Maximum Length: 1
Possible value(s): Y
Notes: This keyword may be used to replace a LOC group if the granule indeed has global coverage.

Keyword: GLOBAL_GRANULES_ONLY
Synopsis: Only global granules should be returned in the result.
Parent Group(s): [INVENTORY_SEARCH]
ODL Type: Symbol
Maximum Length: 1
Possible value(s): Y

Keyword: GRANULE
Synopsis: Collection of metadata about data granule
Parent Group(s): DATASET
Child Group(s): BROWSE_TYPE, GRANULE_ID, PARAMETER, POINT_LOC group, POLYGON_LOC group, PROCESSING_LEVEL, RANGE_LOC group, SENSOR_NAME, SOURCE_NAME, START_DATE, STOP_DATE, [CAMPAIGN], [COMMENT], [DAY_NIGHT], [GLOBAL_GRANULE], [PACKAGE_ID]
ODL Type: Aggregate N/A
Notes:

1. One and only one of the groups or keywords defining spatial coverage of the granule is required.

2. PARAMETER and CAMPAIGN are required if provided in the INVENTORY_SEARCH.
3. If SENSOR_NAME and SOURCE_NAME are not given the DATASET level, SENSOR_NAME and SOURCE_NAME must be given at the GRANULE level.

Keyword: GRANULE_ID
Synopsis: Granule's ID from Inventory
Parent Group(s): BROWSE_REQUEST, GRANULE, IMAGE
ODL Type: String
Maximum Length: 50
Possible value(s): any string

Keyword: GRANULE_LIMIT
Synopsis: Number of granules requested per data set
Parent Group(s): INVENTORY_SEARCH
ODL Type: Integer
Maximum Length: 10
Possible value(s): 1 to 2147483647

Keyword: GUIDE_SEARCH
Synopsis: Collection of criteria for a Guide search
Child Group(s): CAMPAIGN, DATASET_ID, SENSOR_NAME, SOURCE_NAME, PARAMETER, KEYWORD, MONITOR
ODL Type: Aggregate

Keyword: IMAGE
Synopsis: Provides attributes of an image
Parent Group(s): INTEGRATED_BROWSE_RESULT
Child Group(s): DATASET_ID, GRANULE_ID, IMAGE_ID, IMAGE_SIZE
ODL Type: Aggregate

Keyword: IMAGE_ID
Synopsis: Image identifier from Data Center
Parent Group(s): IMAGE group
ODL Type: String
Maximum Length: 30
Possible value(s): any string

Keyword: IMAGE_SIZE
Synopsis: Image size in bytes
Parent Group(s): IMAGE group
ODL Type: String
Maximum Length: 10
Possible value(s): 1 to 2147483647

Keyword: IMS_STAFF
Synopsis: Sent with every client message. Usually blank unless the client was run by a member of the IMS Staff. It comes from the IMS staff environment variable (shell set).

Parent Group(s): VERSION
ODL Type: String

Keyword: INFO_PROMPT
Synopsis: Data Center-supplied string to describe use of 'additional info' on the Order screen.
Parent Group(s): PACKAGE
ODL Type: String
Maximum Length: 80

Keyword: INITIAL_USER_KEY
Synopsis: Set by shell for Data Center hosted clients. Original password used at the Data Center when first registering a user.
Parent Group(s): PRODUCT_REQUEST
ODL Type: String
Maximum Length: 12

Keyword: INTEGRATED_BROWSE_RESULT
Synopsis: Provides result of BROWSE_REQUEST
Child Group(s): DATA_CENTER_ID, IMAGE group, MESSAGE_ID, MONITOR Group, STATUS_CODE
ODL Type: Aggregate

Keyword: INVENTORY_RESULT
Synopsis: Provides result set from inventory query
Child Group(s): DATA_CENTER_ID, MESSAGE_ID, MONITOR group, NUMBER_OF_DATASETS, STATUS_CODE, [DATASET group], [UNMAPPED_FIELD]
ODL Type: Aggregate

Keyword: INVENTORY_SEARCH
Synopsis: Provides data to perform inventory query

Child Group(s): GRANULE_LIMIT, MESSAGE_ID, MONITOR group, [BROWSE_ONLY], [CAMPAIGN], [DATASET_ID], [DAY_NIGHT], GLOBAL_GRANULES_ONLY, [PARAMETER], POINT_LOC group, POLYGON_LOC group, [PROCESSING_LEVEL], RANGE_LOC group, [SENSOR_NAME], [SOURCE_NAME], [START_DATE], [START_DAY_OF_YEAR], [STOP_DATE], [STOP_DAY_OF_YEAR]
ODL Type: Aggregate
Note: One and only one type of spatial coverage is required in the INVENTORY_SEARCH group and at least one of the SOURCE_NAME, SENSOR_NAME, or PARAMETER keywords.

Keyword: KEYWORD
Synopsis: A search string of characters used in Guide.
Parent Group(s): GUIDE_SEARCH
ODL Type: String
Maximum Length: 80

Keyword: LAST_NAME
Synopsis: The user's last name.
Parent Group(s): BILLING_ADDRESS, CONTACT_ADDRESS, SHIPPING_ADDRESS
ODL Type: String
Maximum Length: 20

Keyword: LATITUDE
Synopsis: Latitude for a point on the globe.
Parent Group(s): POINT_LOC, POLYGON_LOC
ODL Type: Sequence Real
Maximum Length: 8
Possible value(s): -90.0000 to +90.0000

Keyword: LATITUDE_DISTANCE
Synopsis: Degrees separating center point and latitude corner point.
Parent Group(s): XHAIRS
ODL Type: String
Maximum Length: 9

Keyword: LINE_ITEM
Synopsis: Information needed for requesting a package.
Parent Group(s): PRODUCT_REQUEST
Child Group(s): EST_COST, MEDIA_TYPE, PACKAGE_ID, PROCESSING_OPTION
ODL Type: Aggregate

Keyword: LONGITUDE
Synopsis: Longitude for a point on the globe.
Parent Group(s): POINT_LOC, POLYGON_LOC
ODL Type: Sequence Real
Maximum Length: 9
Possible value(s): -180.0000 to +180.0000

Keyword: LONGITUDE_DISTANCE
Synopsis: Degrees separating center point and longitude corner point.
Parent Group(s): XHAIRS
ODL Type: String
Maximum Length: 10

Keyword: MAP_PROJECTION_TYPE
Synopsis: Map projection type selected by the user.
Parent Group(s): POLYGON_LOC
ODL Type: String
Maximum Length: 80
Possible value(s): PLATE_CARREE, NORTH_POLAR_STEREOGRAPHIC,
SOUTH_POLAR_STEREOGRAPHIC

Keyword: MD_ENTRY_ID
Synopsis: Global Change Master Directory Entry ID
Parent Group(s): DATASET, GCMD_SEARCH
ODL Type: String
Maximum Length: 31
Possible value(s): any string

Keyword: MEDIA_FORMAT
Synopsis: Media distribution format for delivering selected data.
Parent Group(s): MEDIA_TYPE, LINE_ITEM
ODL Type: String, Aggregate (see note)
Maximum Length: 30, group (see note)
Note: MEDIA_FORMAT is used in two contexts:

1. Under PACKAGE group the values are subgroup names.
2. Under LINE_ITEM, the values are user selected identifying the distribution format for delivery of the data.

Keyword: MEDIA_TYPE
Synopsis: The distribution media for delivering selected data.
Parent Group(s): PROCESSING_OPTIONS
Child Group(s): TYPE-ID, NUMBER-OF-MEDIA-FORMAT
ODL Type: String, Aggregate (see note)
Maximum Length: 20, group (see note)
Note: MEDIA_TYPE is used in two contexts:

1. Under PACKAGE group the values are subgroup names.
2. Under LINE_ITEM, the values are user selected identifying the distribution media for delivery of the data.

Keyword: MESSAGE_ID
Synopsis: Identifier used to track messages.
Parent Group(s): BROWSE_REQUEST, DIRECTORY_RESULT, DIRECTORY_SEARCH, FTP_BROWSE_RESULT, INTEGRATED_BROWSE_RESULT, INVENTORY_RESULT, INVENTORY_SEARCH, PRODUCT_REQUEST, PRODUCT_RESULT
ODL Type: String
Maximum Length: 30
Possible value(s): any string
Note: Generated by Gaea, the IMS client software.

Keyword: MIDDLE_INITIAL
Synopsis: One letter initial for the user's middle name.
Parent Group(s): BILLING_ADDRESS, CONTACT_ADDRESS, SHIPPING_ADDRESS
ODL Type: String
Maximum Length: 1

Keyword: MONITOR
Synopsis: Collection of performance statistics.

Parent Group(s): BROWSE_REQUEST, DIRECTORY_RESULT, DIRECTORY_SEARCH, FTP_BROWSE_RESULT, INTEGRATED_BROWSE_RESULT, INVENTORY_RESULT, INVENTORY_SEARCH, PRODUCT_REQUEST, PRODUCT_RESULT
Child Group(s): RX_CLIENT, RX_SERVER, TX_CLIENT, TX_SERVER
ODL Type: Aggregate
Maximum Length: 84

Keyword: NORTH_LATITUDE
Synopsis: Northern most latitude for an area on the globe.
Parent Group(s): RANGE_LOC
ODL Type: Real
Maximum Length: 8
Possible value(s): -90.0000 to +90.0000

Keyword: NUMBER_OF_DATASETS
Synopsis: Number of data sets included in query result set.
Parent Group(s): DIRECTORY_RESULT, INVENTORY_RESULT
ODL Type: Integer
Maximum Length: 10
Possible value(s): 1 to 2147483647

Keyword: NUMBER_OF_GRANULES
Synopsis: The number of granules included in the package.
Parent Group(s): PACKAGE
ODL Type: Integer
Maximum Length: 10
Possible value(s): 1 to 2147483647

Keyword: NUMBER_OF_GRANULE_HITS
Synopsis: Number of granules from this data set included in query result set.
Parent Group(s): DATASET

ODL Type: Integer
Maximum Length: 10
Possible value(s): 1 to 2147483647

Keyword: NUMBER_OF_MEDIA
Synopsis: Indicates how many media choices are available.
Parent Group(s): PROCESSING_OPTIONS
ODL Type: Integer
Maximum Length: 10
Possible value(s): 1 to 2147483647

Keyword: NUMBER_OF_OPTIONS
Synopsis: Indicates how many processing options are available.
Parent Group(s): PACKAGE
ODL Type: Integer

Maximum Length: 10
Possible value(s): 1 to 2147483647

Keyword: OPTION_ID
Synopsis: The valid value for selected processing options.
Parent Group(s): PROCESSING_OPTIONS
ODL Type: String

\Keyword: ORG_CENTER
Synopsis: Originating center for MD search.
Parent Group(s): DATASET, GCMD_SEARCH
ODL Type: String
Maximum Length: 31

Keyword: ORGANIZATION
Synopsis: Additional address information, e.g., NASA.
Parent Group(s): CONTACT_ADDRESS, DAAC_CONTACT_ADDRESS, BILLING_ADDRESS
ODL Type: String
Maximum Length: 60

Keyword: PACKAGE
Synopsis: The collection of granules or data which can be ordered from an archive.
Parent Group(s): INVENTORY_RESULT
Child Group(s): DATA_CENTER_ID, DATASET_ID PACKAGE_ID, COMMENT, NUMBER_OF_GRANULES, NUMBER_OF_OPTIONS PROCESSING_OPTIONS
ODL Type: String
Notes:

1. OPTION 1: for use when all package information is sent for the whole inventory result and is sent before the first DATASET group (disfavored and may not be implemented).
2. OPTION 2: for use when package information is sent in front of each relevant data set group.
3. OPTION 3: for use when package information is sent within each relevant data set group and before the granule group(s).

Keyword: PACKAGE_CONTACT_ADDRESS
Synopsis: Contact information for the particular package where there are multiple contacts at the Data Center. (Used mostly by international partners.)
Parent Group(s): PRODUCT_RESULT
ODL Type: Aggregate

Keyword: PACKAGE_ID
Synopsis: Names of valid IMS distributed products. If the package information is the same for all granules in the data set and there is one product per granule, then use the character '*' for the PACKAGE_ID.
Parent Group(s): GRANULE, LINE_ITEM, PACKAGE
ODL Type: Sequence String

Maximum Length: 50

Keyword: PACKAGE_SIZE
Synopsis: The size of the package in bytes of data.
Parent Group(s): PROCESSING_OPTIONS
ODL Type: Integer
Maximum Length: 10
Possible value(s): 1 to 2147483647

Keyword: PARAMETER
Synopsis: Valid value that is a geophysical term associated with a data set or granule.
Parent Group(s): DATASET, [DIRECTORY_SEARCH], GRANULE, [INVENTORY_SEARCH]
ODL Type: Sequence String
Maximum Length: 80
Notes:
1. PARAMETER is required in the DATASET or GRANULE groups of the INVENTORY_RESULT group.
2. PARAMETER can be given in the DATASET group if and only if the value of PARAMETER is the same for all the GRANULES in the DATASET group.

Keyword: PHONE
Synopsis: Voice telephone number of associated person.
Parent Group(s): BILLING_ADDRESS, CONTACT_ADDRESS, SHIPPING_ADDRESS
ODL Type: String
Maximum Length: 22
Possible value(s): any string

Keyword: POINT_LOC
Synopsis: Single point on the globe.
Parent Group(s): GRANULE, INVENTORY_SEARCH
Child Group(s): LATITUDE, LONGITUDE
ODL Type: Aggregate

Keyword: POLE_INCLUDED
Synopsis: Pole is included in described search area.
Parent Group(s): POLYGON_LOC
ODL Type: Symbol
Maximum Length: 1
Possible value(s): N,| S
Notes: If not included in the location group then no pole included in region.

Keyword: POLYGON_LOC
Synopsis: Group of four latitude longitude pairs describing an area on the globe.
Parent Group(s): GRANULE, INVENTORY_SEARCH
Child Group(s): LATITUDE, LONGITUDE, [POLE_INCLUDED], MAP_PROJECTION_TYPE, TANGENT_LATITUDE, TANGENT_LONGITUDE

ODL Type: Aggregate

Keyword: PONG
Synopsis: Message for testing server availability and responsiveness
Child Group(s): MONITOR, VERSION
ODL Type: Aggregate

Keyword: PROCESSING_LEVEL
Synopsis: Level to which data has been processed.
Parent Group(s): [GRANULE], [DATASET], [DIRECTORY_SEARCH], [INVENTORY_SEARCH]
ODL Type: Symbol or Sequence String
Maximum Length: 2
Possible value(s): 0, 1, 1a, 1b, 2, 3, 4
Note: DATASET unique, currently under review

Keyword: PROCESSING_OPTIONS
Synopsis: User requested processing of GRANULE to produce a product.
Parent Group(s): LINE_ITEM PACKAGE
ODL Type: Sequence String
Maximum Length: 30

Keyword: PRODUCT_REQUEST
Synopsis: Provides data for product request.
Child Group(s): BILLING_ADDRESS group, CONTACT_ADDRESS group, DATA_CENTER_ID, LINE_ITEM group, MESSAGE_ID, MONITOR group, REQUEST_ID, SHIPPING_ADDRESS group, USER_ID, USER_AFFILIATION group
ODL Type: Aggregate

Keyword: PRODUCT_RESULT
Synopsis: Group of information including Data Center contact information acknowledging a product request.
Child Group(s): CONTACT_ADDRESS group, DATA_CENTER_ID, MESSAGE_ID, MONITOR group, STATUS_CODE
ODL Type: Aggregate

Keyword: PROTOCOL_VERSION
Synopsis: Version of message passing protocol, e.g., 3.5.
Parent Group(s): VERSION
ODL Type: Real

Keyword: QUIT
Synopsis: Termination message.
Parent Group(s): MONITOR
ODL Type: Aggregate

Keyword: RANGE_LOC
Synopsis: Group of maximum and minimum latitudes and longitudes describing an area.
Parent Group(s): [DIRECTORY_SEARCH], GRANULE, INVENTORY_SEARCH
Child Group(s): EAST_LONGITUDE, NORTH_LATITUDE, SOUTH_LATITUDE, WEST_LONGITUDE
ODL Type: Aggregate

Keyword: RESTRICTION
Synopsis: Details of any ordering restrictions placed on the data set.
Parent Group(s): DATASET group
ODL Type: Sequence String
Maximum Length: 60
Possible value(s): any string

Keyword: REQUEST_ID
Synopsis: ID, assigned by the client software, to a product request for tracking. Generated by Gaea, the IMS client software.
Parent Group(s): PRODUCT_REQUEST
ODL Type: String
Maximum Length: 30

Keyword: RX_CLIENT
Synopsis: Time stamp when the client received the entire ODL message
Parent Group(s): MONITOR group
ODL Type: Sequence STRING
Maximum Length: 20
Possible value(s): two part: seconds (required), microseconds (optional)
Notes: integer number of seconds as returned by the time () call or the gettimeofday call

Keyword: RX_SERVER
Synopsis: Time stamp when the server received the entire ODL message
Parent Group(s): MONITOR group
ODL Type: Sequence STRING
Maximum Length: 20
Possible value(s): two part: seconds (required), microseconds (optional)
Notes: integer number of seconds as returned by the time () call or the gettimeofday call

Keyword: SENDER_VERSION
Synopsis: Descriptor identifying the name and number of the sender (client or server) that sent the message.
Parent Group(s): VERSION
ODL Type: String
Maximum Length: 16

Keyword: SENSOR_NAME
Synopsis: Name(s) of sensor.
Parent Group(s): GRANULE, DATASET, [DIRECTORY_SEARCH], [INVENTORY_SEARCH]
ODL Type: Sequence String
Maximum Length: 30

Keyword: SERVER_VERSION
Synopsis: Optional descriptor identifying the server version, and is stored in the group = VERSION.
Parent Group(s): VERSION
ODL Type: String
Maximum Length: 16

Keyword: SHIPPING_ADDRESS
Synopsis: Address where requested data is to be sent.
Parent Group(s): PRODUCT_REQUEST
Child Group(s): CITY, EMAIL, FAX, FIRST_NAME, MIDDLE_INITIAL, LAST_NAME, PHONE, STATE, COUNTRY, STREET, ZIP
ODL Type: Aggregate

Keyword: SESSION_ID
Synopsis: The session ID is a character string used to uniquely identify a user session. It is composed of the following fields, separated by colon (:) characters:

1. The Fully Qualified Domain Name (FQDN) of the host on which the client is running, e.g., "killians.gsfc.nasa.gov"
2. The process ID (PID) of the client. The PID is an integer typically in the range 1-30,000.
3. The data of the session start, in YYMMDD format.
4. The time of the session start, in HHMMSS format.

Parent Group(s): VERSION
ODL Type: Character string
Maximum Length: 87

Keyword: SOURCE_NAME
Synopsis: Name(s) of source/platform.
Parent Group(s): GRANULE, [DIRECTORY_SEARCH], [INVENTORY_SEARCH]

ODL Type: Sequence String
Maximum Length: 30

Keyword: SOUTH_LATITUDE
Synopsis: Southern most latitude for an area on the globe
Parent Group(s): RANGE_LOC
ODL Type: Real
Maximum Length: 8
Possible value(s): -90.0000 to +90.0000

Keyword: START_DATE
Synopsis: Beginning of temporal interest
Parent Group(s): GRANULE, [DIRECTORY_SEARCH], [INVENTORY_SEARCH]
ODL Type: Date
Maximum Length: 20
Possible value(s): yyyy-mm-ddThh:mm:ss | yyyy-mm-ddThh:mm:ssZ

Keyword: START_DAY_OF_YEAR
Synopsis: Beginning day of seasonal interest
Parent Group(s): [INVENTORY_SEARCH]
ODL Type: Integer
Maximum Length: 3
Possible value(s): 1 TO 366

Keyword: STATE
Synopsis: US Postal state abbreviation for associated person
Parent Group(s): BILLING_ADDRESS, CONTACT_ADDRESS, SHIPPING_ADDRESS
ODL Type: String
Maximum Length: 20
Possible value(s): any string

Keyword: STATUS_CODE
Synopsis: Numeric code giving status of query and/or server
Parent Group(s): DIRECTORY_RESULT, FTP_BROWSE_RESULT, INTEGRATED_BROWSE_RESULT, INVENTORY_RESULT, PRODUCT_RESULT, QUIT
ODL Type: Integer
Maximum Length: 4
Possible value(s): 1 to 20, or 1000
Notes:

01 successful query; query results returned
02 no match found
03 data for selected source are not archived at DAAC
04 data for selected sensor are not archived at DAAC
05 data set is not archived at DAAC
06 data for selected parameter(s) not archived at DAAC
07 data for selected source, sensor, parameter(s) and/or data set are not archived at DAAC
08 pertinent inventory system unavailable, try again later
09 bad message; message contains syntax error(s)
10 requested function not supported by this DAAC
11 system error, please try again later
12 search too broad, narrow spatial and/or temporal search criteria
13 no data for selected campaign archived at DAAC, please reconstruct Search Query
14 browse_granules_only selected, but no granules having browse data match
15 global_granules_only selected, but no granules having global coverage match
16 no data for requested processing level at this DAAC, please reconstruct Search Query
17 bad message; protocol error
18 system busy; try again later
19 system error; contact user support
20 data not found due to spatial and/or temporal limitation
1000 user-requested abort of search

Keyword: STATUS_CODE_COMMENT
Synopsis: Data Center provided commentary related to status code for communications.
Parent Group(s): DIRECTORY_RESULT, FTP_BROWSE_RESULT, INTEGRATED_BROWSE_RESULT, INVENTORY_RESULT, DIRECTORY_RESULT, FTP_BROWSE_RESULT, INTEGRATED_BROWSE_RESULT
ODL Type: sequence string
Maximum Length: 128

Keyword: STATS_REPORT
Synopsis: Statistics message corresponding to a session start, session end or product request report message.
Child Group(s): MESSAGE_ID, STATS_REPORT_LENGTH, STATS_REPORT_STRING, MONITOR, VERSION

ODL Type: Aggregate

Keyword: STATS_REPORT_LENGTH
Synopsis: The length of a STATS_REPORT_STRING string
Parent Group(s): STATS_REPORT
ODL Type: Integer
Maximum Length: 0 to maximum size of integer on host

Keyword: STATS_REPORT_STRING
Synopsis: The statistics report string
Parent Group(s): STATS_REPORT
ODL Type: String
Maximum Length: Variable; up to at least 15,000 characters

Keyword: STOP_DATE
Synopsis: Date terminating interval of temporal interest.
Parent Group(s): GRANULE, [DIRECTORY_SEARCH], [INVENTORY_SEARCH]
ODL Type: Date
Maximum Length: 20
Possible value(s): yyyy-mm-ddThh:mm:ss | yyyy-mm-ddThh:mm:ssZ

Keyword: STOP_DAY_OF_YEAR
Synopsis: Ending day of seasonal interest.
Parent Group(s): [INVENTORY_SEARCH]
ODL Type: Date
Maximum Length: 3
Possible value(s): 1 to 366

Keyword: TANGENT_LATITUDE
Synopsis: Current tangent (center) latitude of projection map.
Parent Group(s): POLYGON_LOC
ODL Type: Real
Maximum Length: 8
Possible value(s): -90.0000 to +90.0000

Keyword: TANGENT_LONGITUDE
Synopsis: Current tangent (center) latitude of projection map.
Parent Group(s): POLYGON_LOC
ODL Type: Real
Maximum Length: 9
Possible value(s): -180.0000 to +180.0000

Keyword: TITLE
Synopsis: Part of the User Profile. A user's formal designation.
Parent Group(s): CONTACT_ADDRESS, SHIPPING_ADDRESS, BILLING_ADDRESS
ODL Type: String
Maximum Length: 5

Keyword: TOTAL_FILE_SIZE
Synopsis: Combined uncompressed byte size of all FTP requests (may be exact or approximated).
Parent Group(s): FTP_BROWSE_RESULT
ODL Type: String
Maximum Length: 10
Possible value(s): 1 to 2147483647

Keyword: TX_CLIENT
Synopsis: Time stamp when client transmitted entire ODL message.
Parent Group(s): MONITOR group
ODL Type: Sequence STRING
Maximum Length: 20
Possible value(s): two part: seconds (required), microseconds (optional)
Notes: integer number of seconds as returned by the time () call or the gettimeofday call

Keyword: TX_SERVER
Synopsis: Time stamp when server transmitted entire ODL message.
Parent Group(s): MONITOR group
ODL Type: Sequence STRING
Maximum Length: 20
Possible value(s): two part: seconds (required), microseconds (optional)
Notes: integer number of seconds as returned by the time () call or the gettimeofday call

Keyword: TYPE
Synopsis: Affiliation categories: Government, Commercial, Academic, Other.
Parent Group(s): USER_AFFILIATION
ODL Type: String
Maximum Length: 15

Keyword: TYPE_ID
Synopsis: The valid values for selected media types.
Parent Group(s): MEDIA_TYPE
ODL Type: String
Maximum Length: 30

Keyword: UNMAPPED_FIELD
Synopsis: Field(s) given in query not used in inventory search.
Parent Group(s): [INVENTORY_RESULT]
ODL Type: Sequence String
Maximum Length:
Possible value(s): any keyword contained in the INVENTORY_SEARCH group

Keyword: USER_AFFILIATION
Synopsis: General information for user services statistics.
Parent Group(s): PRODUCT_REQUEST
Child Group(s): CATEGORY, TYPE

ODL Type: Aggregate

Keyword: VALID_ACCOUNTS
Synopsis: Contains DAAC provided valid account information associated with a particular data set. Is an optional or a repeating group.
Parent Group(s): DATASET

ODL Type: Group
Notes:

1. There may be 0 valid account groups sent in inventory/dataset group.
2. If the user has no valid account, then 1 valid account group will be sent containing only the error object with information to instruct or inform the user.
3. For cases with multiple accounts, many valid accounts groups will be sent, each containing mandatory account number with optional balance and error fields.

Keyword: WEST_LONGITUDE
Synopsis: Western most longitude for an area on the globe.
Parent Group(s): RANGE_LOC
ODL Type: Real
Maximum Length: 9
Possible value(s): -180.0000 to +180.0000

Keyword: ZIP
Synopsis: US Postal ZIP code for associated person.
Parent Group(s): BILLING_ADDRESS, CONTACT_ADDRESS, SHIPPING_ADDRESS
ODL Type: String
Maximum Length: 15
Possible value(s): any string

Appendix A. Dependent Valids Format

A.1 Valids Support File (Bitmap) Processing

Periodically, it is necessary for the DAACs to update the existing set of dependent valids which reside with the EOSDIS V0 IMS Client, ECS Release A Search and Order Tool and the V0 Gateway. To this end, the DAACs submit, to the V0/ECS Science Team (a system-level team consisting of both V0 and ECS personnel), valids information (e.g., all valid search fields for each of the DAACs' data sets) in the form of flat ASCII files using ODL syntax. Using the "New Valids Ingest Software," the "DAACs ODL Valids Input File" is compared against a "Valids Master File" maintained by the V0/ECS Science Team. Any new valids are recorded and passed to the V0/ECS Science Team, which checks for consistency between the new and existing terminology. and negotiates/resolves any inconsistencies. Any necessary changes are made to the DAACs ODL Valids Input File and to the Valids Master File. Then the "New Valids Ingest Software" is re-run, combining all DAAC ODL Valids inputs to produce the "Valids Support Files" (bitmap files) containing dependencies that are ultimately sent out by the Support File Server (sf_server). The Dependent Valids Support File Processing is depicted in Figure A-1, and is discussed in further detail in the paragraphs which follow.

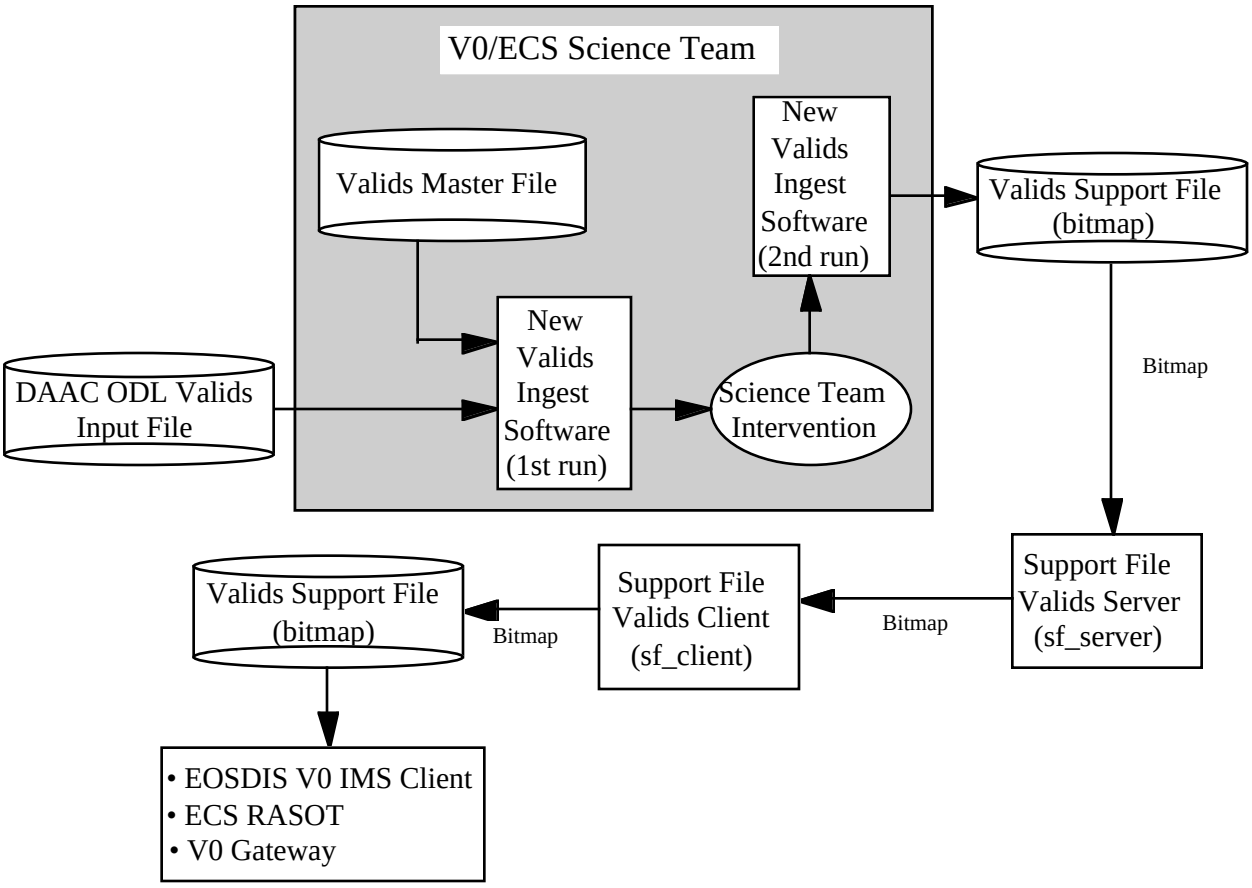


Figure A-1. Valids Support File (Bitmap) Processing

A.1.1 Input to New Valids Ingest Software

The inputs to the New Valids Ingest Software shall be the DAAC ODL Valids Input Files, and a Valids Master File containing a list of valid strings---this file is maintained by the V0/ECS Science Team. In a separate file is the filter list.

A.1.1.1 Valids Master File

The Valids Master File shall have the following format for each line:

Number, 1 space, KEYWORD=, VALID STRING, {ALIAS1} ... {ALIASn}, newline character "\n",

where

- Number represents the valid index number (i.e., bit position of valid)
- KEYWORD represents the name of the field (i.e., the field type)---currently, there are 7 field types, including the following:

  - DATA_CENTER_ID
  - DATASET_ID
  - SOURCE
  - SENSOR
  - PARAMETER
  - CAMPAIGN
  - PROCESSING_LEVEL

- VALID STRING represents the actual name of the valid (e.g., GSFC, MODIS, TRMM, etc.). There shall be no space after the "=" symbol and before the VALID STRING---i.e., KEYWORD=VALID STRING (for example, SENSOR=MODIS). The VALID STRING can be up to 80 characters in length.
- ALIAS names are alternate names by which the VALID STRING can be referrred to. ALIAS names, by convention, are always enclosed in braces {} and are limited to 80 characters each (not including the braces).
- The lines of the Input Master List are sorted in numerical order---i.e., sorted by valid index number.

An example of the format of the Valids Master File is depicted in Figure A-2.

Also, it should be noted that in the list for each field other than DATA_CENTER_ID or DATASET_ID is a VALID STRING that is: *unspecified*. An example *unspecified* VALID STRING is depicted in Figure A-3.

Number	1 Space	KEYWORD=	VALID STRING	{ALIAS1}	{ALIAS2}	\n
5	1 Space	SENSOR	SAGE-II	{SAGE-2}	{SAGE_II}	\n

EXAMPLE ONLY **EXAMPLE ONLY**

Figure A-2. Example Format of Valids Master File

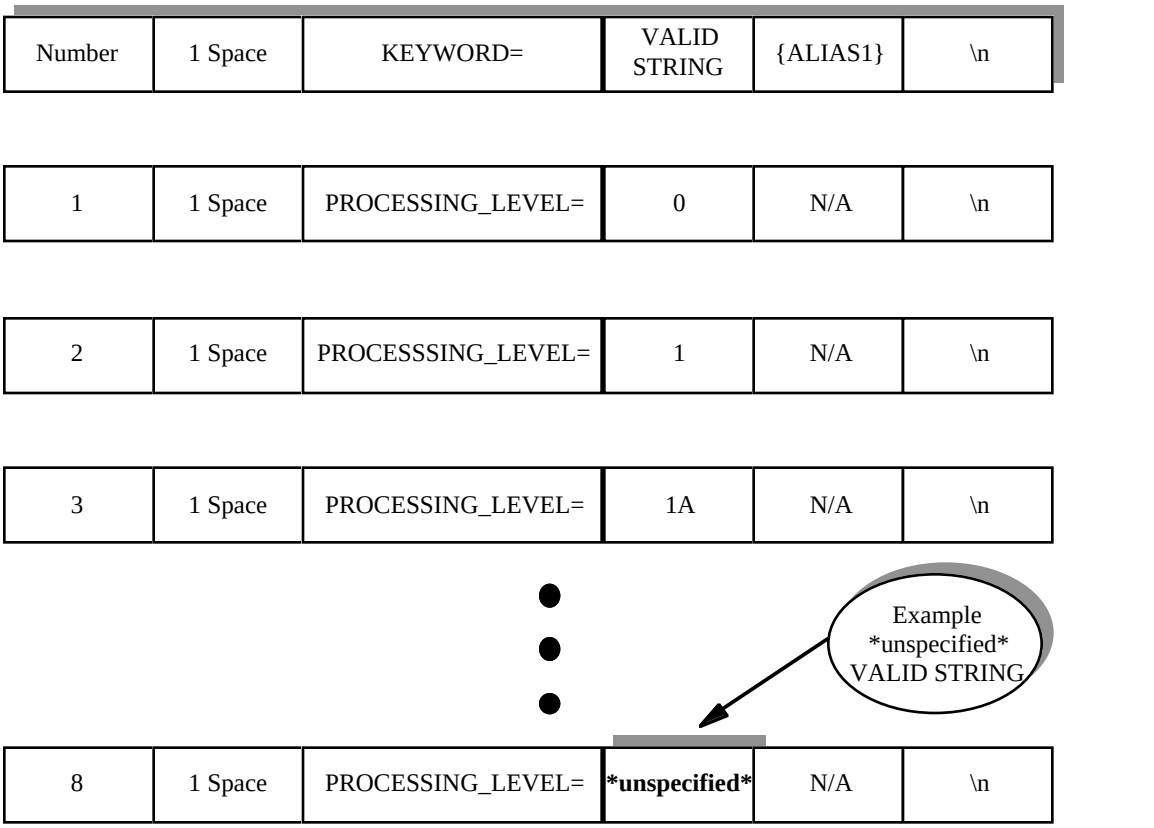


Figure A-3. Example *unspecified* VALID STRING

A.1.1.2 Filter List

In order to help the user formulate queries to the IMS, filters identify meaningful groupings of valid strings into given categories. The filter list has the following format:

KEYWORD=VALID STRING n1, n2, n3 ...

where

- KEYWORD represents the name of the field (i.e., the field type).
- VALID STRING represents the name of the filter.
  - n1, n2, n3 ... represents the list of index numbers of the strings in the Valids Master File. The numbers are separated by commas. A series (or range) of numbers is represented by separating the beginning and ending numbers with a dash; e.g., the series 5,6,7,8 would be represented as 5-8.

An example filter list is provided in Figure A-4.

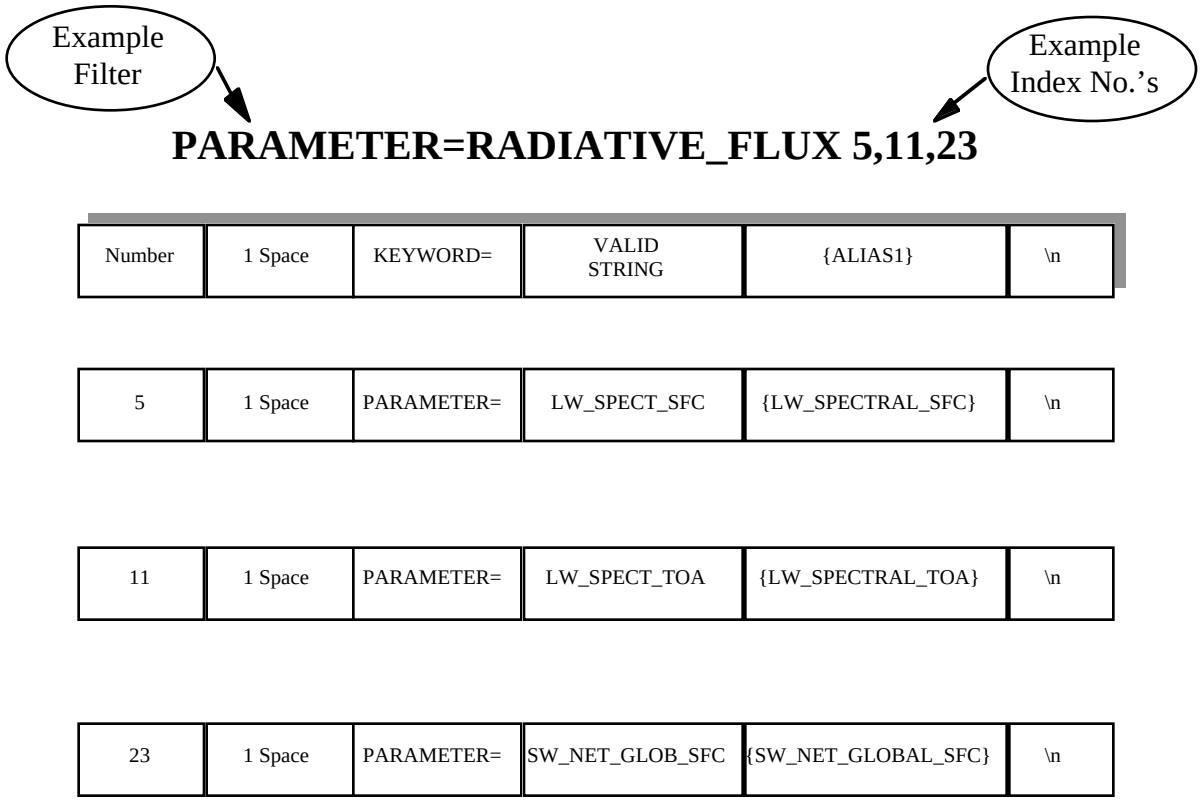


Figure A-4. Example Filter List

A.1.2 New Valids Ingest Software Processing

The New Valids Ingest Software shall process files as follows:

a. The New Valids Ingest Software shall read and process the input master list file. Within the input Valids Master File:

(1) If there is a duplicate index number with the same KEYWORD and a different VALID STRING value, print the occurrences of the duplicates, and abort before writing the bitmap.

(2) If there is a duplicate index number with a different KEYWORD, print the duplicate and abort before writing the bitmap.

(3) If there are two different numbers with the same KEYWORD=VALID STRING, print both lines, and abort before writing the bitmap.

(4) A different number with a different KEYWORD but same VALID STRING is okay.

(5) If there is a duplicate index number with the same KEYWORD and the same VALID STRING value, print the occurrences of the duplicates, and abort before writing the bitmap.

b. The New Valids Ingest Software shall use the master list file to create the new list of valid strings and the field bit mask.

c. The New Valids Ingest Software shall read the valids file from each DAAC.

d. The New Valids Ingest Software shall set the bitmap appropriately. It is recommended that the software ingest at least a complete dependency group before attempting to set the bitmap. In many cases the dependency group is a DATASET GROUP. For each valid string that is found in the dependency group, a bit shall be set in the row for that valid in the column for all the valids in the dependency group. These bits shall also be set in the row and column that represent all of the fields of that dependency group. If a field is not set within a group, the bit for *unspecified* shall be set. The search for the string index shall be made with a case-blind search.

e. The New Valids Ingest Software shall read the filter list to create: a list of filter strings, a filter bitmap, and a filter field bit mask.

A.1.3 Output From New Valids Ingest Processing

The New Valids Ingest Processing shall produce the Valids Support File (bitmap) consisting of the following file(s):

a. bitmap file (including bitmap header)

b. field bit mask file (including bitmap header)

c. filter bitmap file (including bitmap header)

d. filter field bit mask file (including bitmap header)

e. valid string list

f. filter string list

A.1.3.1 Bitmap Header

The bitmap header will be a character string of 80 characters where the last character is a newline character ("\n"). The bitmap header will include, at a minimum, the following items of information:

a. a header version number

b. one (1) space

c. a serial number

d. one (1) space

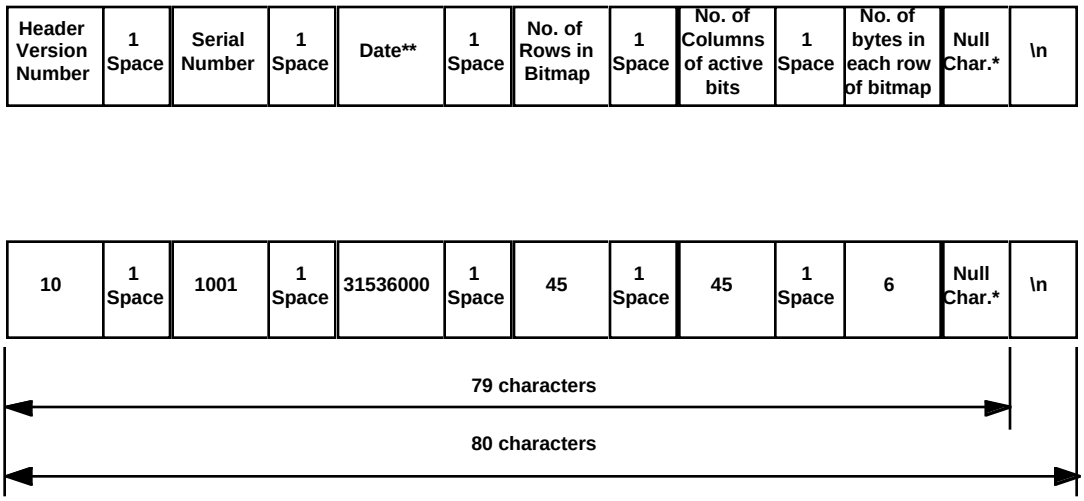
e. date

f. one (1) space

g. number of rows in the bitmap
h. one (1) space
i. number of columns of active bits
j. one (1) space
k. number of bytes in each row of the bitmap
l. NULL characters to pad to 79 characters
m. the newline character ("\n")

Here, "date" represents the time when the New Valids Ingest Software wrote the file (in seconds since 00:00:00 UTC, January 1, 1970). This same header format will be used for the Bitmap File, Field Bit Mask File, Filter Bitmap File, and Filter Field Bit Mask File discussed, respectively, in sections A.1.2.2 through A.1.2.5. An example bitmap header is depicted in Figure A-5.

EXAMPLE ONLY **EXAMPLE ONLY**



*** NULL characters pad to 79 characters**
**** Represents time in seconds since 00:00:00 UTC, Jan 1, 1970**

EXAMPLE ONLY **EXAMPLE ONLY**

Figure A-5. Example Bitmap Header

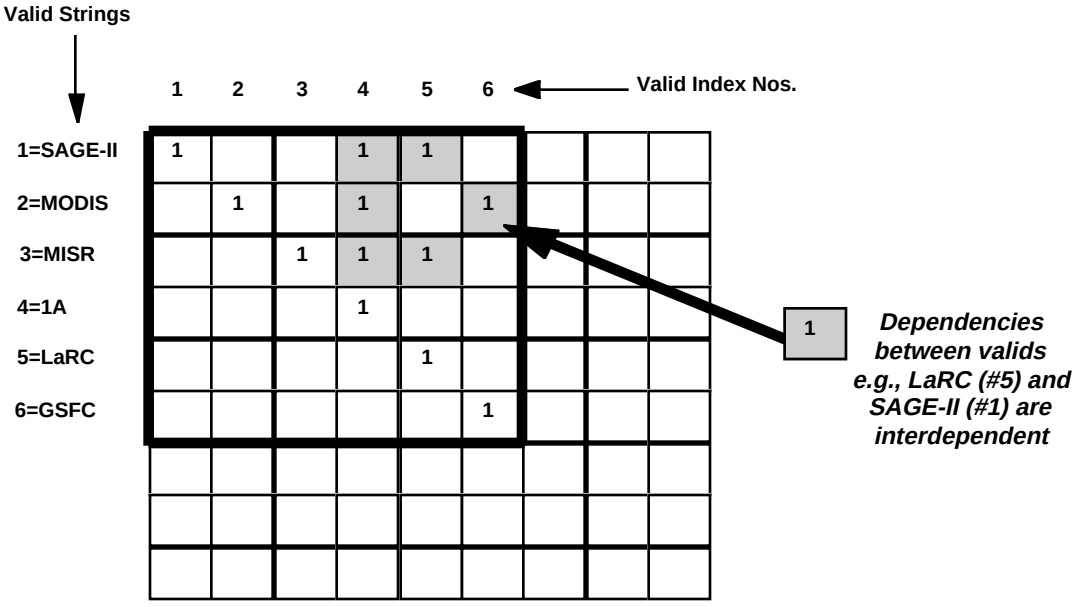
A.1.3.2 Bitmap File

The bitmap file contains a bitmap header and the bitmap data. Within the header, the "Number of Rows in Bitmap" and the "Number of Columns of Active Bits" are the same and they represent the total number of valids. The "Number of Bytes in Each Row of Bitmap" represents the

number of bytes needed to store the number of columns of bits. The bitmap represents the dependencies between the valids. Each row represents a valid and its index is the same as in the master list of valid strings. A bit is "on" when the valid bit column number is compatible with the current valid row. The order of bytes within the bitmap is: a row of bytes followed by the next row of bytes. The order of bits within a byte is the "zeroeth" bit starts on the left. Some bits in the last byte of the row may not be used if the number of bits is not divisible by the number of bits per byte. If the bits are not used they shall be set to zero. An example bitmap file is depicted in Figure A-6.

EXAMPLE ONLY **EXAMPLE ONLY**

Header Version Number	1 Space	Serial Number	1 Space	Date**	1 Space	No. of Rows in Bitmap	1 Space	No. of Columns of active bits	1 Space	No. of bytes in each row of bitmap	Null Char.*	\n
10	1 Space	1001	1 Space	31536000	1 Space	6	1 Space	6	1 Space	1	Null Char.*	\n



EXAMPLE ONLY **EXAMPLE ONLY**

Figure A-6. Example Bitmap File

A.1.3.3 Field Bit Mask File

The field bit mask file contains a bitmap header and the field bit mask data. In this case within the header, the number of rows is set to the number of field types. Currently, the number of field types is seven (7). The number of columns represents the total number of valids. The number of bytes represents the number of bytes needed to store the number of columns of bits. The field bit mask indicates which field each valid represents. Each row of the field bit mask represents one of the 7 field types. A bit is "on" for the valid bit column number, when the valid belongs to the

field represented by the row. The order of bytes within the field bit mask is: a row of bytes followed by the next row of bytes. The order of the rows is as follows: DATA_CENTER_ID, DATASET_ID, SOURCE, SENSOR, PARAMETER, CAMPAIGN, and PROCESSING_LEVEL. The order of bits within a byte is the "zeroeth" bit starts on the left. Some bits in the last byte of the row may not be used if the number of bits is not divisible by the number of bits per byte. If the bits are not used they shall be set to zero. An example field bit mask file is depicted in Figure A-7.

EXAMPLE ONLY **EXAMPLE ONLY**

Header Version Number	1 Space	Serial Number	1 Space	Date**	1 Space	No. of Rows in Bitmap	1 Space	No. of Columns of active bits	1 Space	No. of bytes in each row of bitmap	Null Char.*	\n
10	1 Space	1001	1 Space	31536000	1 Space	7	1 Space	11	1 Space	2	Null Char.*	\n

Field Types

Valid Index Nos.

	1	2	3	4	5	6	7	8	9	10	11
DATA_CENTER_ID	1										1
DATASET_ID			1					1			
SOURCE					1						
SENSOR						1					
PARAMETER				1			1				
CAMPAIGN									1		
PROCESSING_LEVEL		1								1	

EXAMPLE ONLY **EXAMPLE ONLY**

Figure A-7. Example Field Bit Mask File

A.1.3.4 Filter Bitmap File

The filter bitmap file contains a bitmap header and the filter bitmap data. The number of rows represents the number of filters. The number of bit columns represents the total number of valids. The number of bytes represents the number of bytes required to store the total number of columns of bits. Each row represents a filter. The bits are "on" for those columns that represent the valids for that filter. The order of bytes within the bitmap is: a row of bytes followed by the next row of bytes. The order of bits within a byte is the "zeroeth" bit starts on the left. Some bits in the last byte of the row may not be used if the number of bits is not divisible by the number of

bits per byte. If the bits are not used they shall be set to zero. An example filter bitmap file is depicted in Figure A-8.

EXAMPLE ONLY **EXAMPLE ONLY**

Header Version Number	1 Space	Serial Number	1 Space	Date**	1 Space	No. of Rows in Bitmap	1 Space	No. of Columns of active bits	1 Space	No. of bytes in each row of bitmap	Null Char.*	\n
10	1 Space	1001	1 Space	31536000	1 Space	6	1 Space	12	1 Space	2	Null Char.*	\n

Filters — Valid Index Nos.

Filters	1	2	3	4	5	6	7	8	9	10	11	12		
DATA_CENTER_ID=DAAC 1,2	1	1												
DATASET_ID=CLOUDS 3,4			1	1										
SOURCE=EDOS 5,6					1	1								
SENSOR=RADIOMETER 7,8							1	1						
PARAMETER=AIR_QUALITY 9.10									1	1				
CAMPAIGN=RAD_BUDGET 11,12											1	1		
PROCESSING_LEVEL=MAPS 13,14													1	1

EXAMPLE ONLY **EXAMPLE ONLY**

Figure A-8. Example Filter Bitmap File

A.1.3.5 Filter Field Bit Mask File

The filter field bit mask file contains a bitmap header and the filter field bit mask data. In this case within the header, the number of rows is set to the number of field types. Currently, the number of field types is seven (7). The number of columns represents the total number of filters. The number of bytes represents the number of bytes needed to store the number of columns of bits. The field bit mask indicates which field each filter represents. Each row of the field bit mask represents one of the 7 field types. A bit is "on" for the filter bit column number, when the filter belongs to the field represented by the row. The order of bytes within the field bit mask is: a row of bytes followed by the next row of bytes. The order of the rows is: DATA_CENTER_ID, DATASET_ID, SOURCE, SENSOR, PARAMETER, CAMPAIGN, and PROCESSING_LEVEL. The order of bits within a byte is the "zeroeth" bit starts on the left. Some bits in the last byte of the row may not be used if the number of bits is not divisible by the number of bits per byte. If the bits are not used they shall be set to zero. An example filter field bit mask file is depicted in Figure A-9.

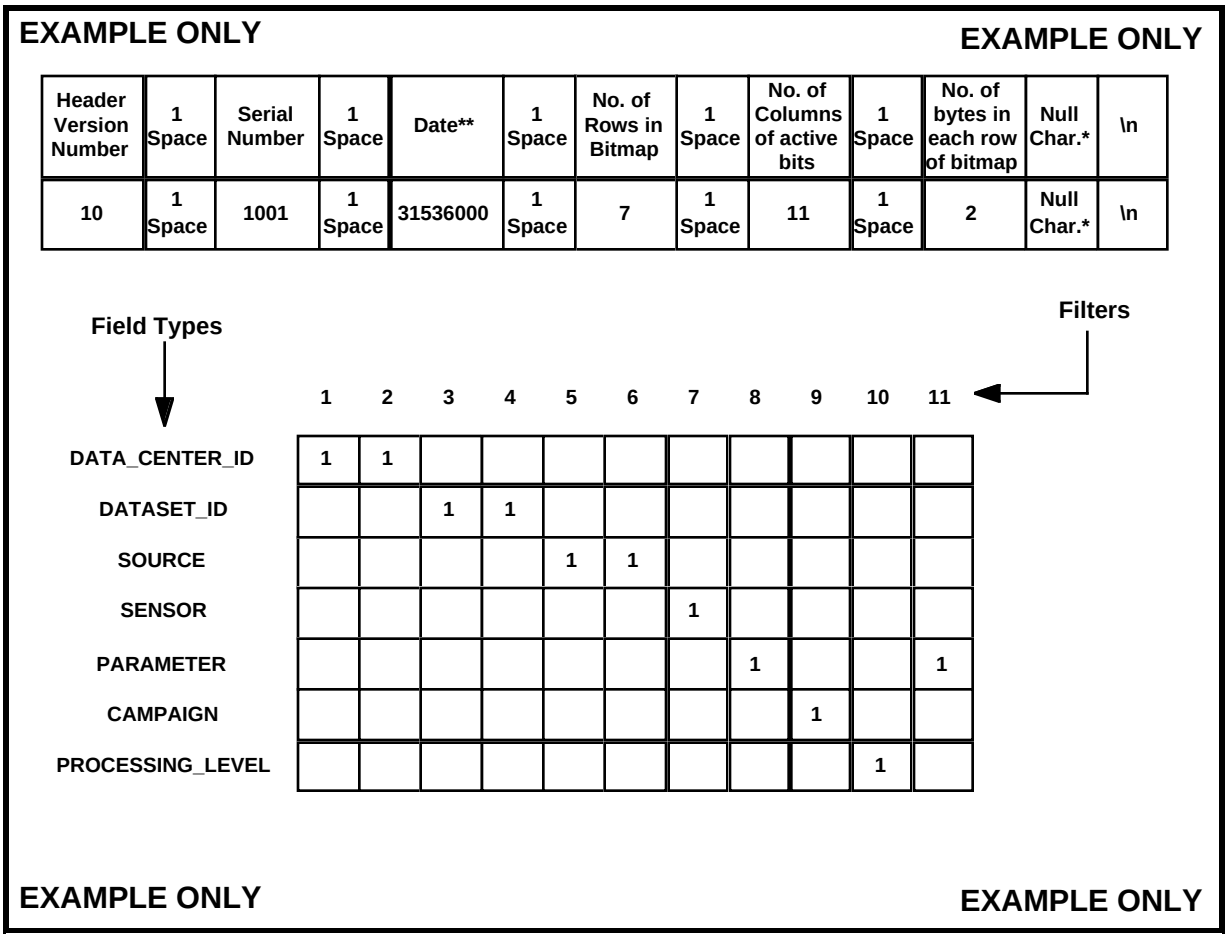


Figure A-9. Example Field Bit Mask File

A.1.3.6 Valid String List

The Valid String List begins with a string list header which contains the following:

a. header version number (character string)
b. one (1) space
c. serial number of the string list (character string)
d. one (1) space
e. UNIX date in seconds from 1/1/70 00:00:00 UTC (character string)
f. a newline character (\n)

The Valid String List contains the list of strings of all valid types. There shall be one valid string (with its aliases) per line. Each string is terminated with a newline character ("\n"). Missing valids shall be indicated by a single newline character. It shall be guaranteed that this list is sorted by Valids Index Number. An example Valid String List is depicted in Figure A-10.

Header Version No.	1 Space	Serial No. of String List	1 Space	UNIX date	\n
10	1 Space	1001	1 Space	31200095	\n

VALID STRING	{ALIAS1}
GSFC	{Goddard}
EP/TOMS Data	{TOMS Ozone}
Earth Probe	{EP}
⋮	
1A	N/A

EXAMPLE ONLY **EXAMPLE ONLY**

Figure A-10. Example Valid String List

A.1.3.7 Filter String List

The Filter String List begins with a string list header. The string list header contains the following:

a. header version number in a character string
b. one (1) space
c. serial number of the string list in a character string
d. one (1) space
e. UNIX date in seconds from 1/1/70 00:00:00 UTC in a character string
f. a newline character (\n)

The Filter String List contains the list of strings consisting of all filter names. There shall be one filter string per line. Each string is terminated with a newline character ("\n"). Missing filters shall be indicated by a single newline character. This list is sorted by filter index order. An example filter string list is depicted in Figure A-11.

Header Version No.	1 Space	Serial No. of String List	1 Space	UNIX date	\n
10	1 Space	1001	1 Space	31200095	\n

FILTER_TYPE=	FILTER STRING
SENSOR=	RADIOMETER 1,2
SENSOR=	POLARIMETER 3,4
CAMPAIGN=	RAD_BUDGET 7,8
⋮	
PROCESSING_LEVEL=	FORECASTS 31,32

EXAMPLE ONLY **EXAMPLE ONLY**

Figure A-11. Example Filter String List

A.1.4 Passing New Valids Support File From Server To Client

Periodically, the Support File Client communicates with the Support File Server to determine whether an updated version of the New Valids Support File is available. If an updated version of the New Valids Support Files exist, they are automatically transferred to the Support File Client in the format of a compressed tar file. The Support File Client does not uncompress or untar these files---it only handles transferring the New Valids Support Files to the EOSDIS V0 IMS Client, ECS RASOT, or (in the Case of Release B), to the V0 Gateway. A script called "Autoxfer" compliments the Support File Client program by performing the untar and handling failed server response.

This page intentionally left blank.

Abbreviations and Acronyms

CCB	Configuration Control Board
CCR	Configuration Change Request
CDRL	Contract Data Requirements List
CSMS	Communications and System Management Segment
DAAC	Distributed Active Archive Center
DBMS	Data Base Management System
DCN	Document Change Notice
ECS	EOSDIS Core System
EOS	Earth Observing System
EOSDIS	Earth Observing System Data and Information System
ESDIS	Earth Science Data and Information System
ESQL	Earth Science Query Language (For ECS)
ESST	Earth Science Search Tool
FQDN	Fully Qualified Domain Name
FTP	File Transfer Protocol
GCMD	Global Change Master Directory
GUI	Graphical User Interface
HDF	Hierarchical Data Format
HTML	HyperText Markup Language
HTTP	Hypertext Transport Protocol
I&T	integration and test
ICD	Interface Control Document
ID	Identifier
IDL	Interactive Data Language
IK	IMS Kernel
IMS	Information Management System

IP	Internet Protocol
Ir-1	interim release -1
IRD	Interface Requirements Document
NCSA	National Center for Supercomputer Applications
ODL	Object Description Language
OODCE	Object Oriented Distributed Computing Environment
PID	Process Identifier
RASOT	Release A Search and Order Tool
TBS	To Be Supplied
TRMM	Tropical Rainfall Measuring Mission (joint US-Japan)
V0	Version 0
V1	Version 1
WAIS	Wide Area Information Server
WWW	World-Wide Web